

Figure 1

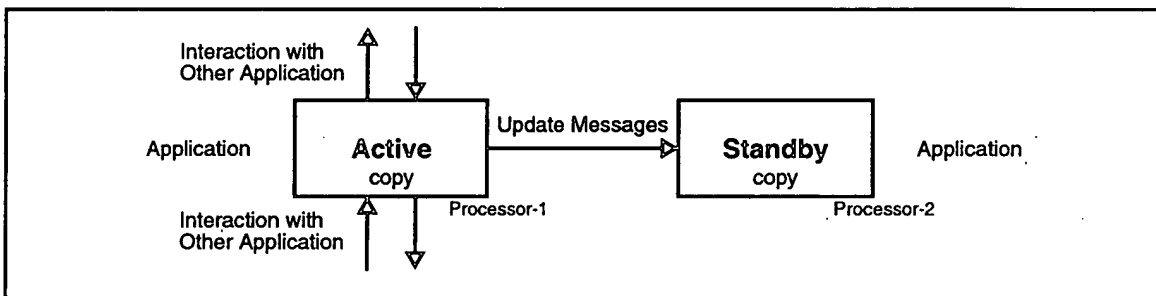


Figure 2

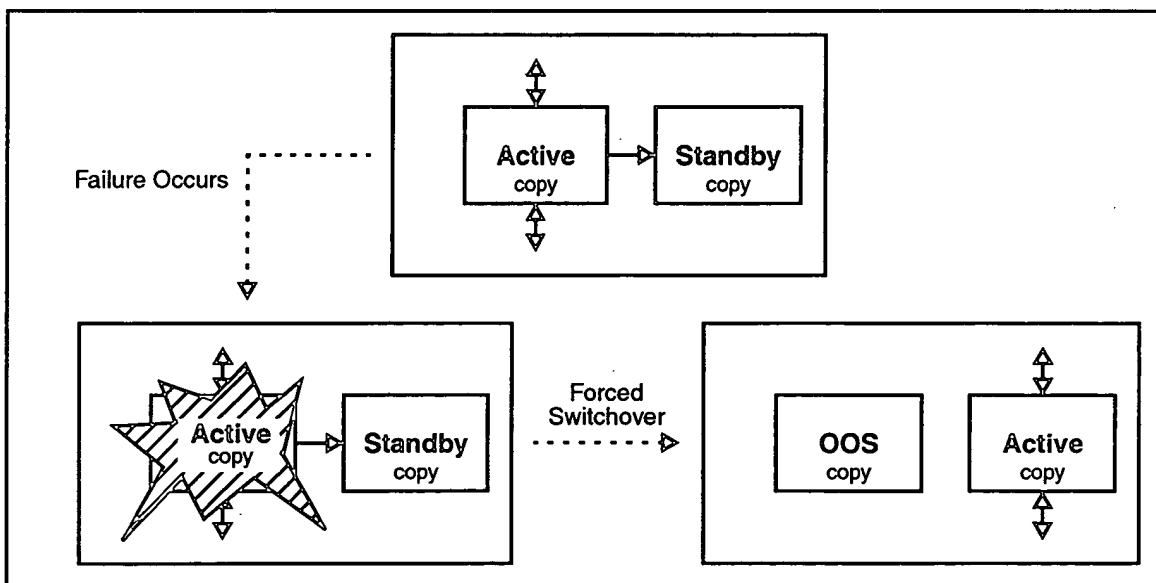


Figure 3

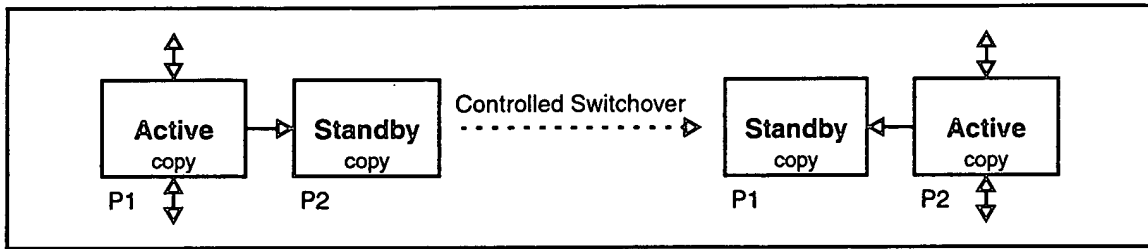


Figure 4

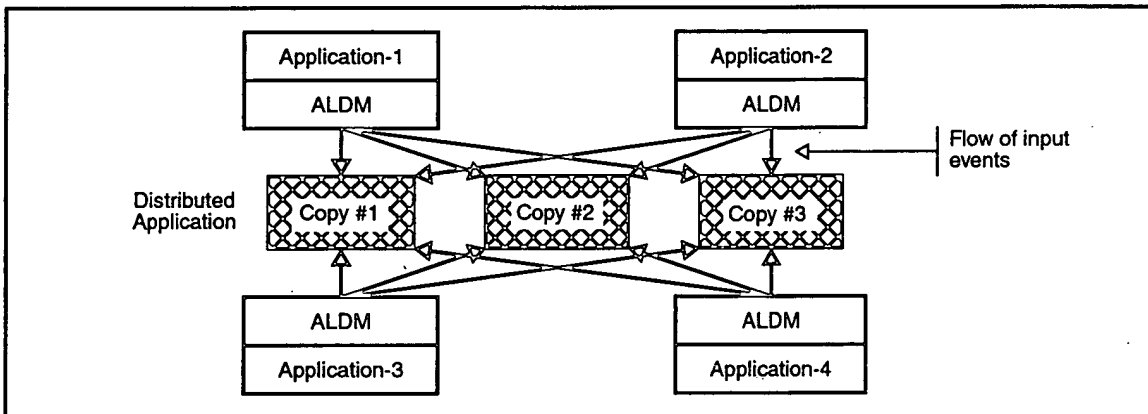


Figure 5

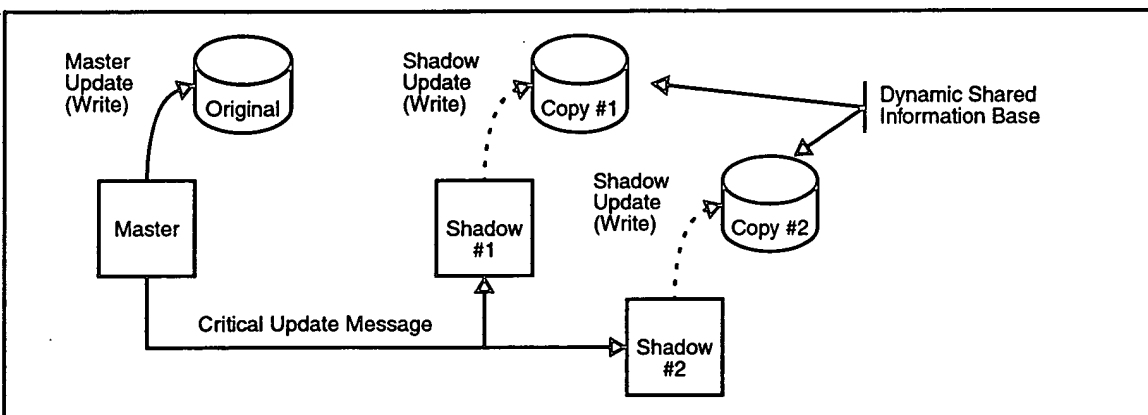


Figure 6

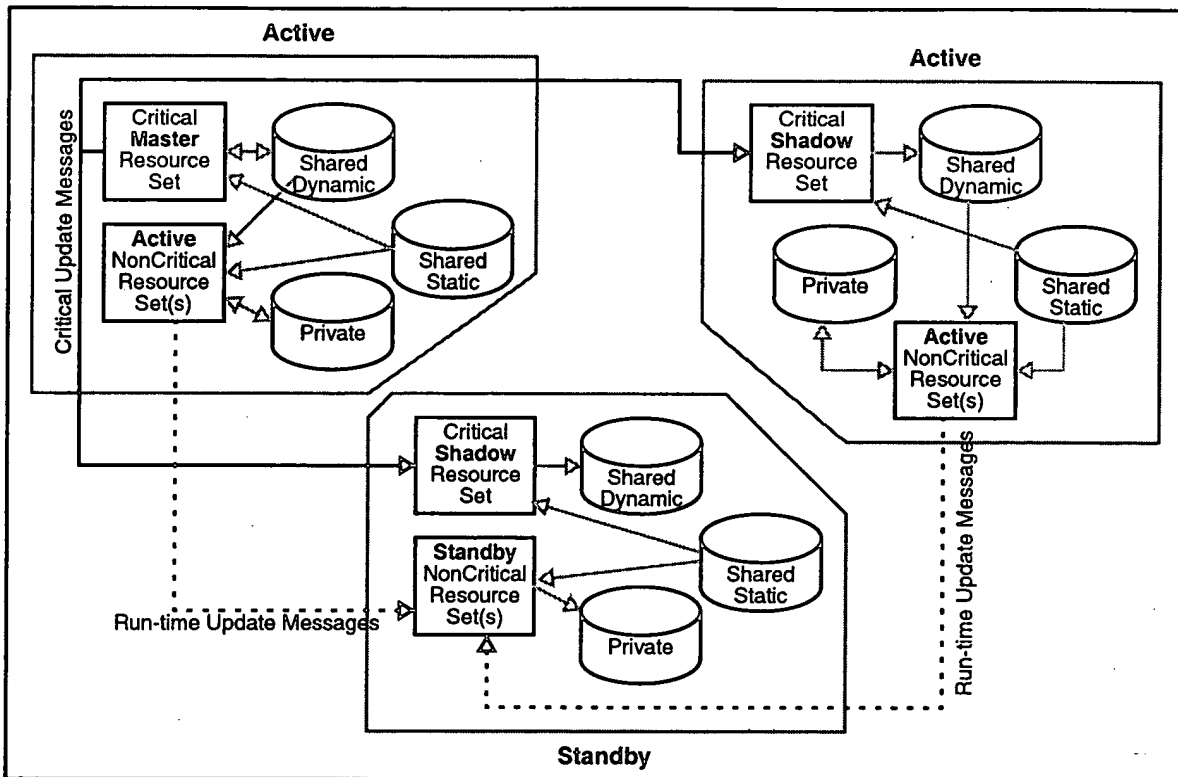


Figure 7

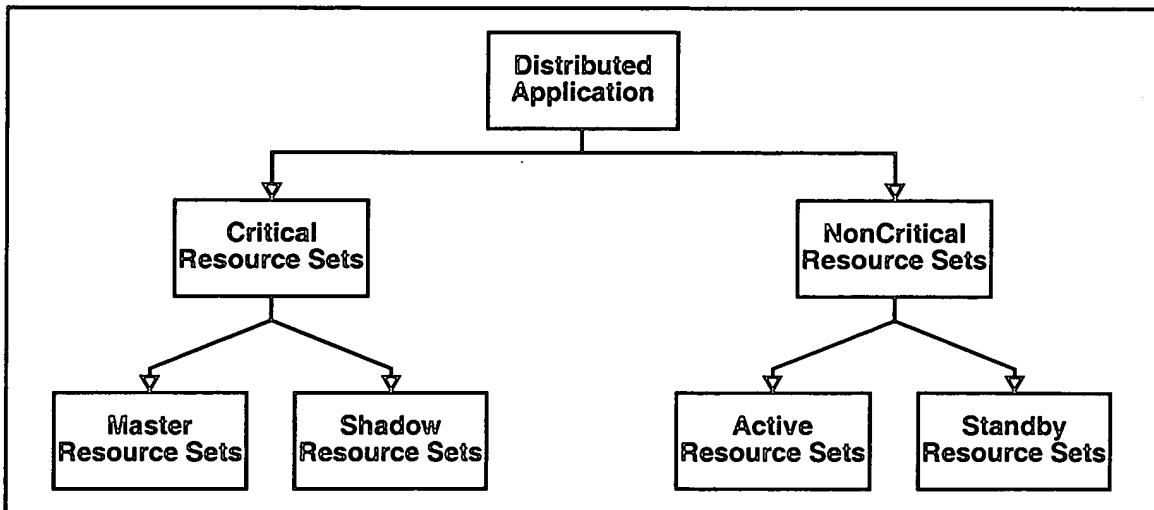


Figure 8

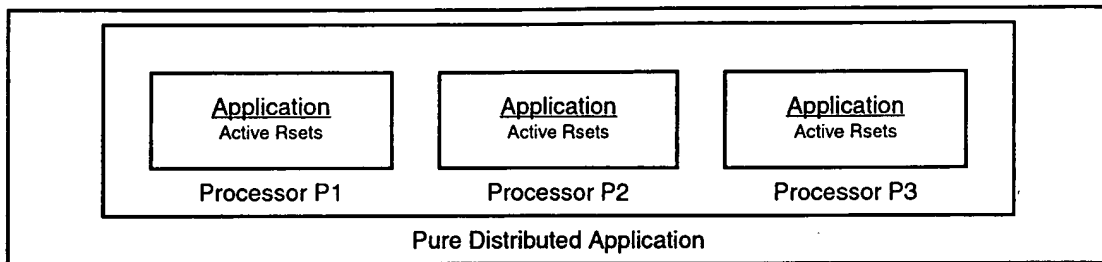


Figure 9

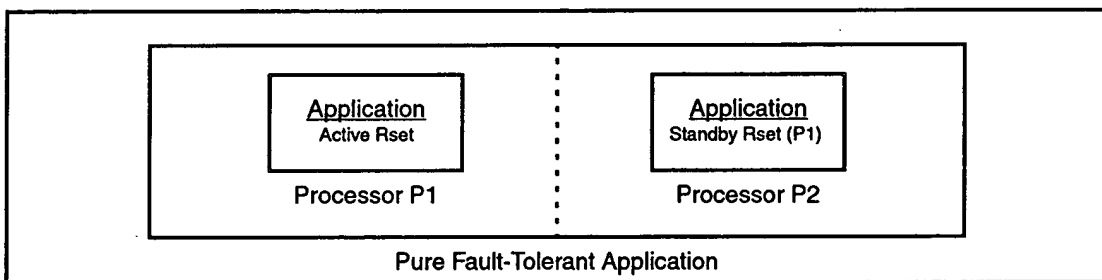


Figure 10

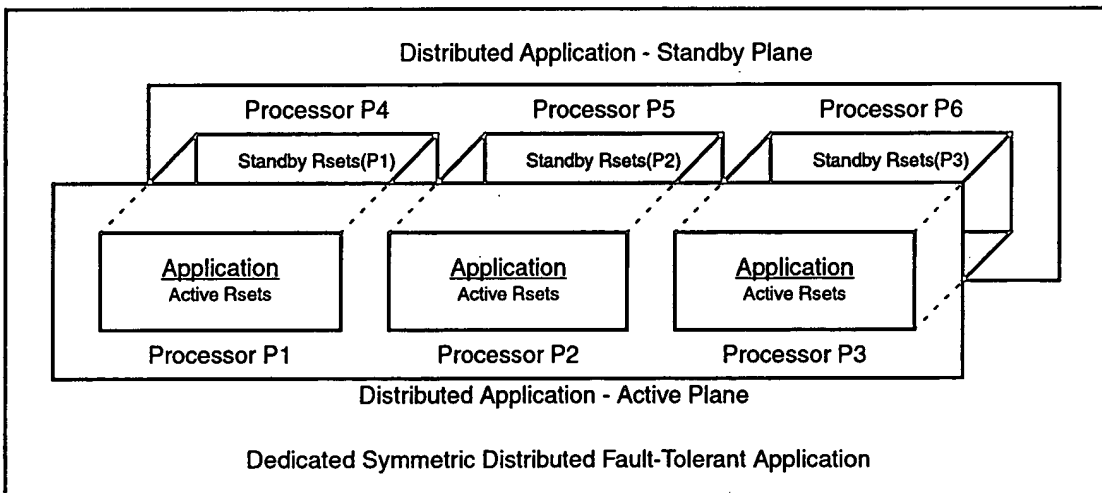


Figure 11

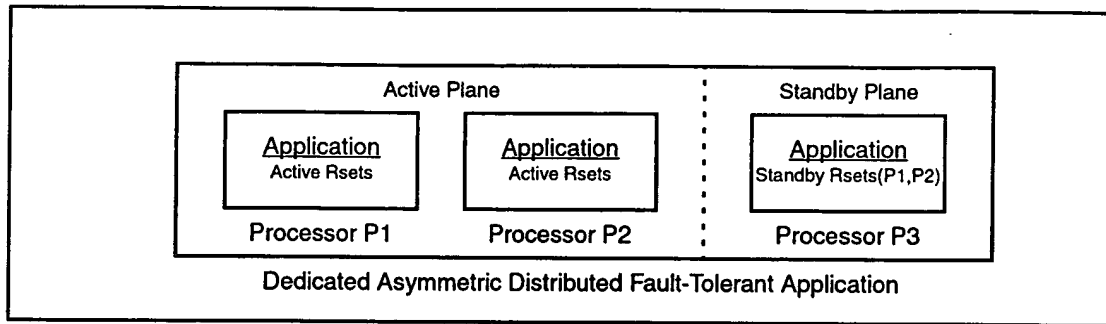


Figure 12

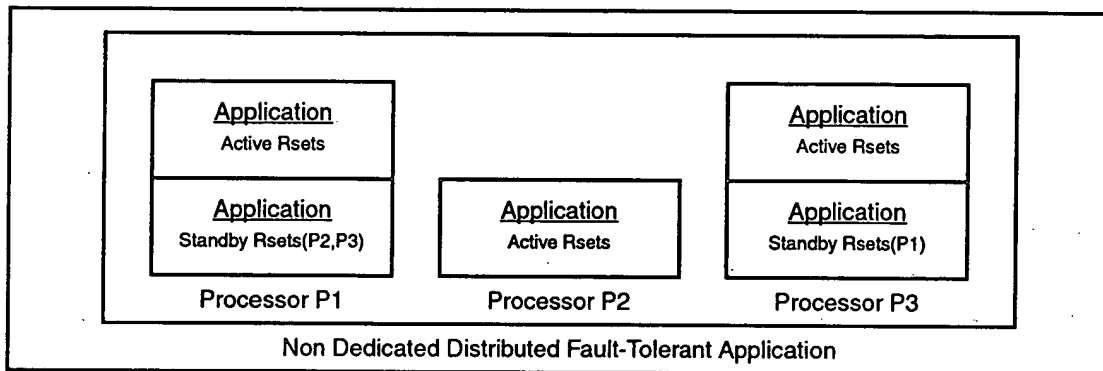


Figure 13

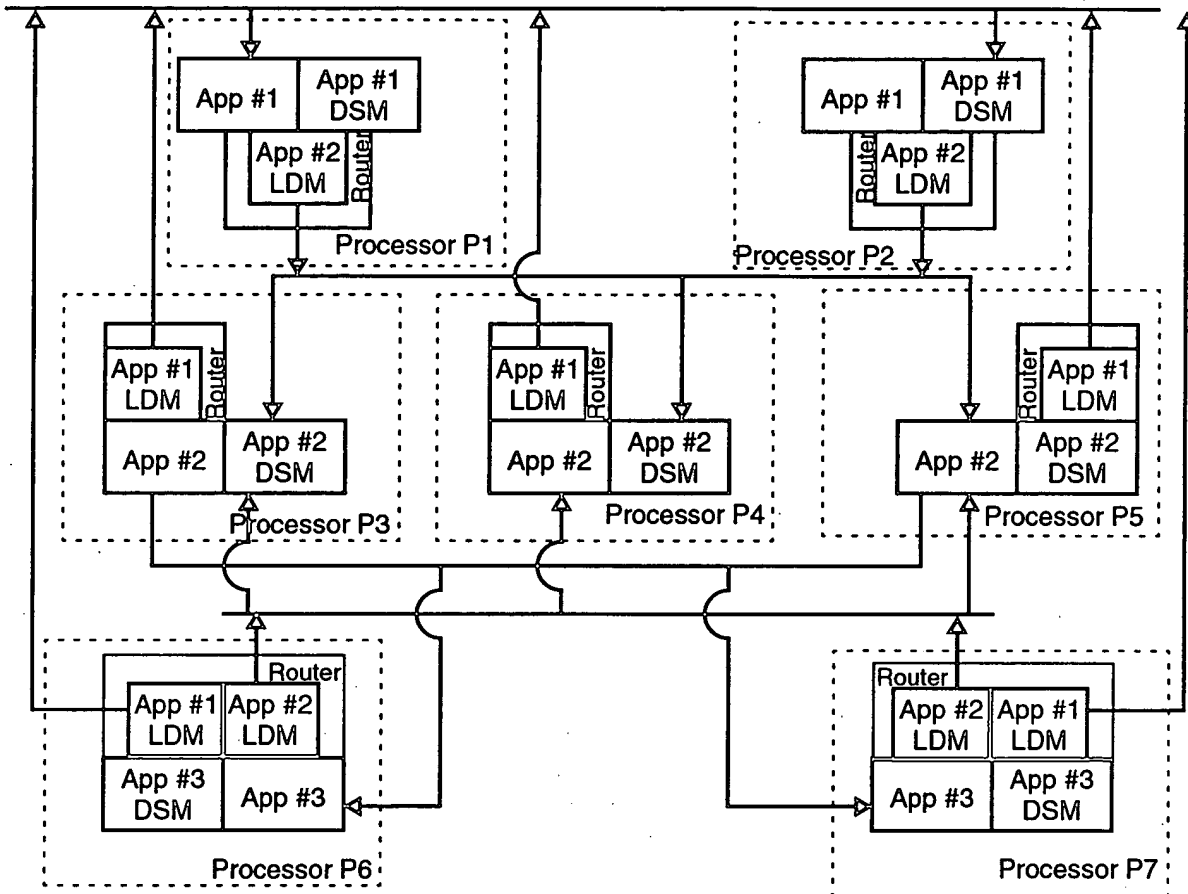


Figure 15

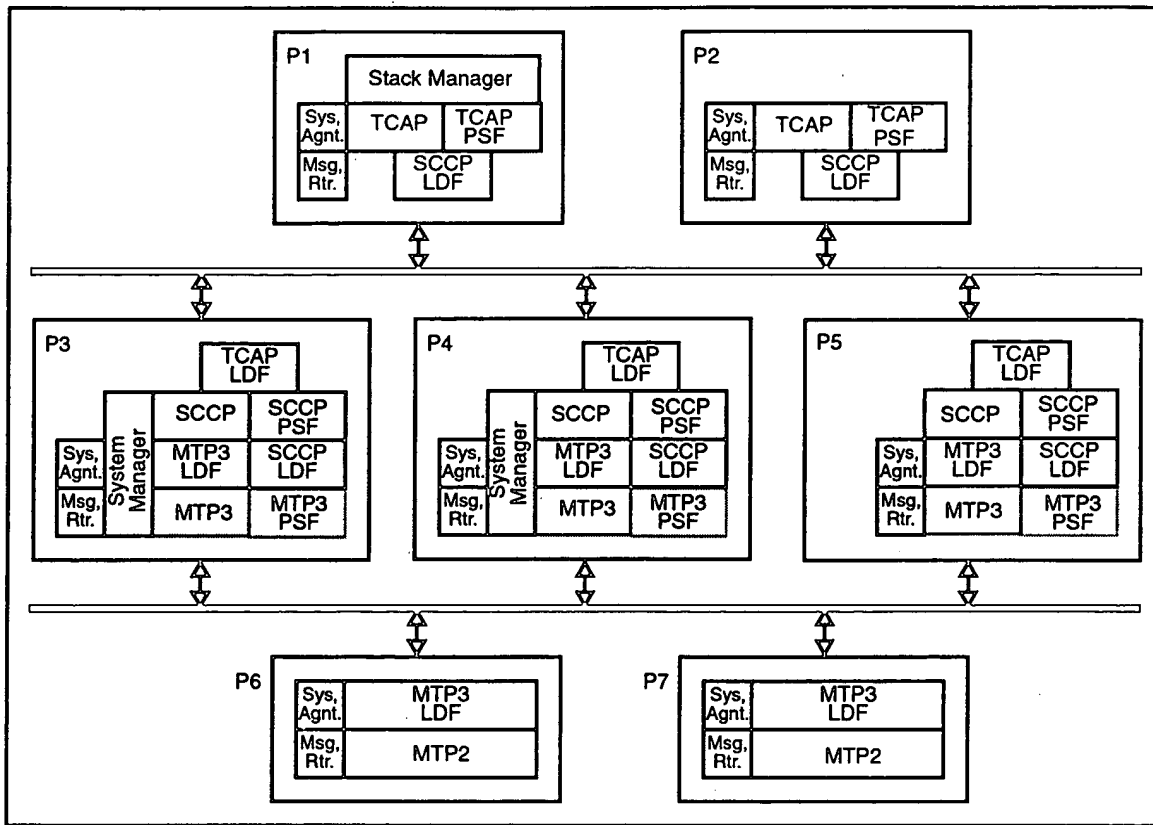


Figure 16

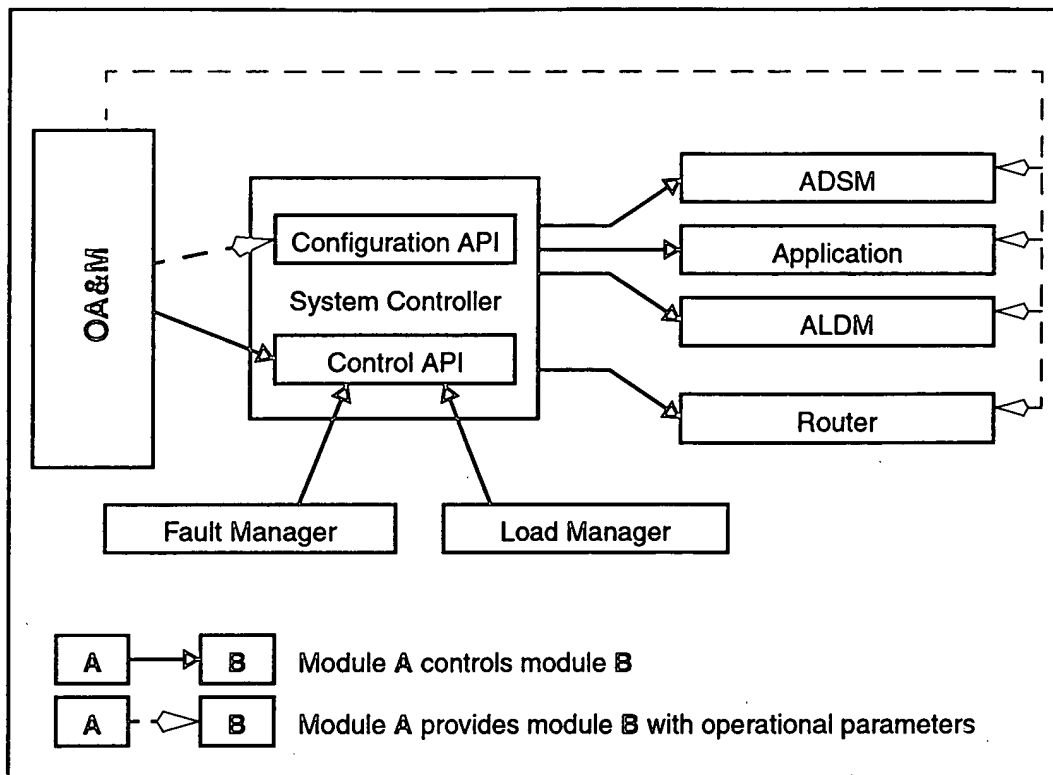


Figure 17

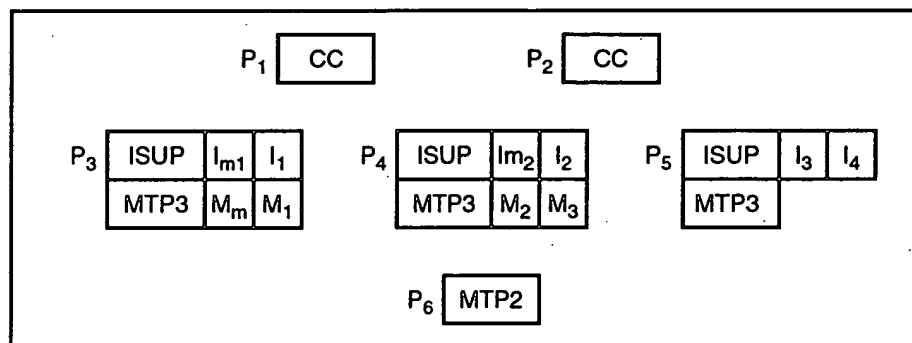


Figure 18

I_{m1} : ISUP management critical resource set

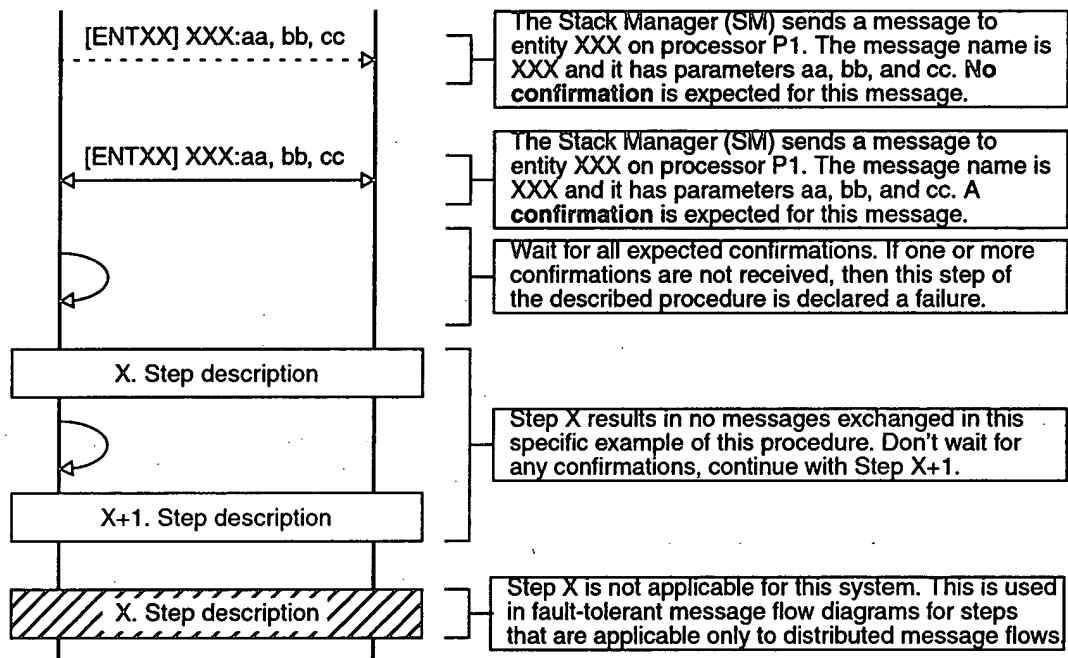
I_{m2} : ISUP other critical resource set

I_1, I_2, I_3, I_4 : ISUP non critical resource sets

M_m : MTP3 management critical resource set

M_1, M_2, M_3 : MTP3 non critical resource sets

M_1, M_2, M_3 : MTP3 non critical resource sets



The Stack Manager (SM) sends a message to entity XXX on processor P1. The message name is XXX and it has parameters aa, bb, and cc. **No confirmation** is expected for this message.

The Stack Manager (SM) sends a message to entity XXX on processor P1. The message name is XXX and it has parameters aa, bb, and cc. A **confirmation** is expected for this message.

Wait for all expected confirmations. If one or more confirmations are not received, then this step of the described procedure is declared a failure.

X. Step description

Step X results in no messages exchanged in this specific example of this procedure. Don't wait for any confirmations, continue with Step X+1.

X+1. Step description

Step X is not applicable for this system. This is used in fault-tolerant message flow diagrams for steps that are applicable only to distributed message flows.

Figure 19

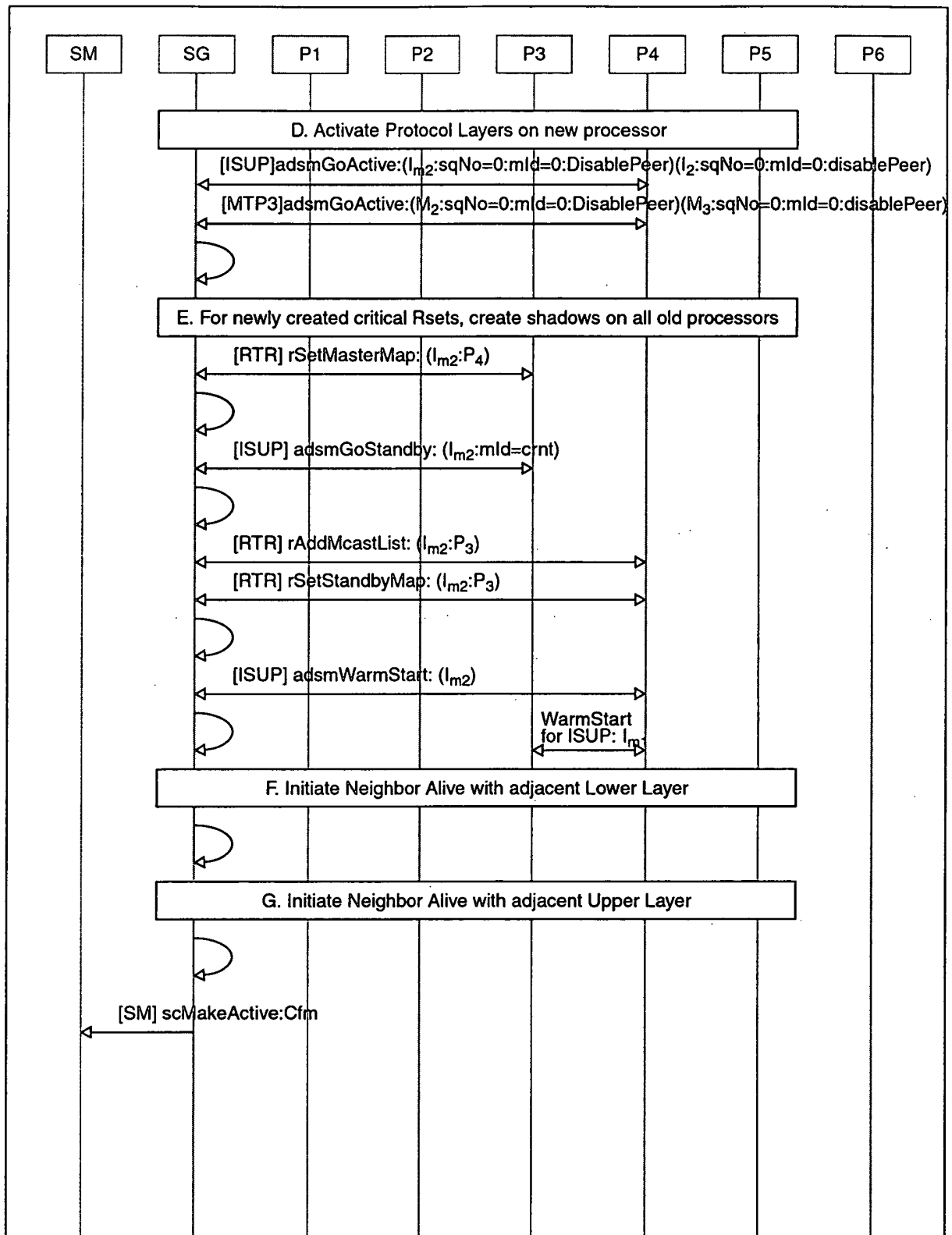


Figure 22

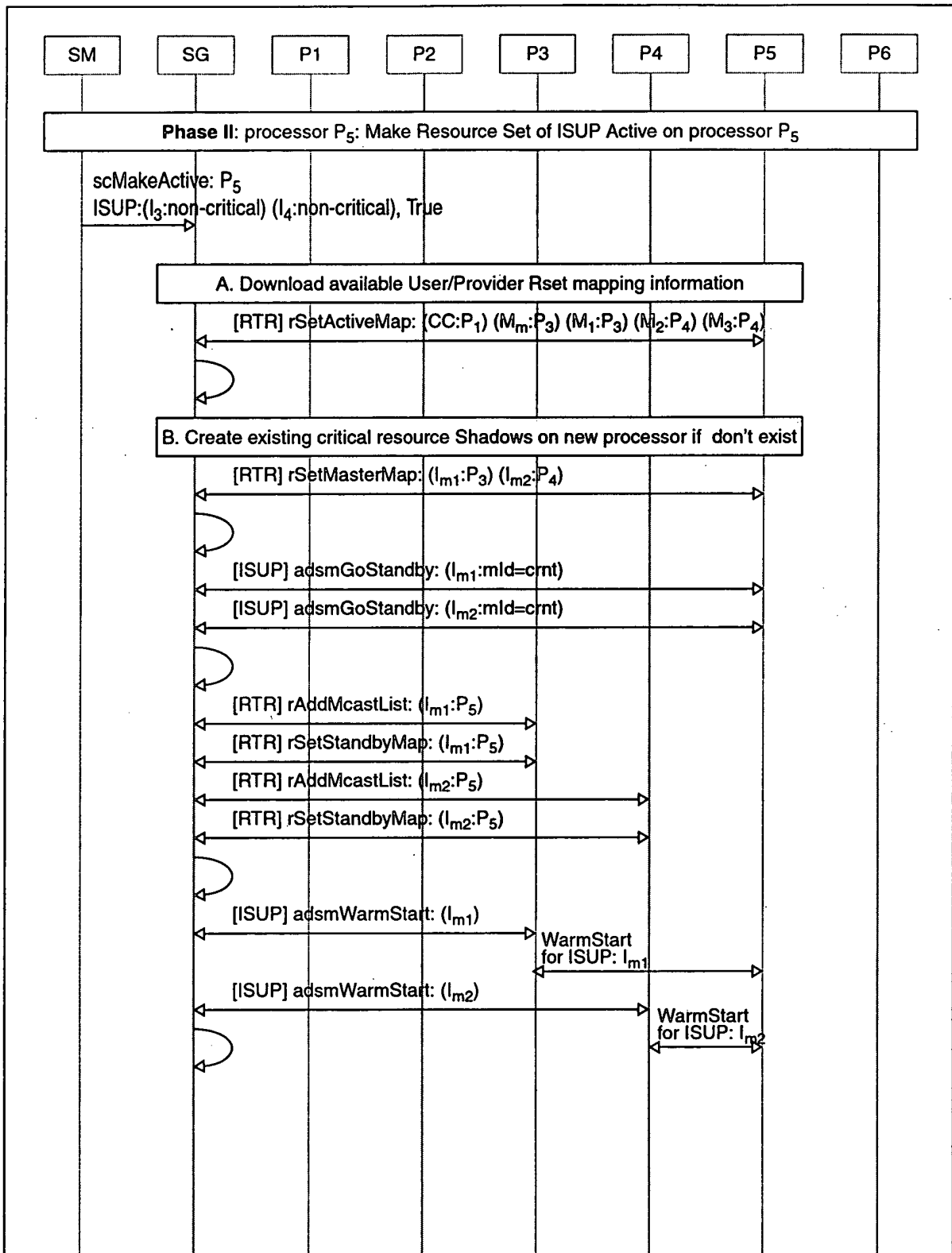


Figure 23

```

sequenceDiagram
    participant SM
    participant SG
    participant P1
    participant P2
    participant P3
    participant P4
    participant P5
    participant P6

    Note over SM,SG,P1,P2,P3,P4,P5,P6: C. Download new mappings to adjacent routers
    SG->>P1:[RTR] rSetActiveMap: (I3:P5) (I4:P5)
    SG->>P2:[RTR] rSetActiveMap: (I3:P5) (I4:P5)
    SG->>P3:[RTR] rSetActiveMap: (I3:P5) (I4:P5)
    SG->>P4:[RTR] rSetActiveMap: (I3:P5) (I4:P5)
    SG->>P5:[RTR] rSetActiveMap: (I3:P5) (I4:P5)
    SG->>P6:[RTR] rSetActiveMap: (I3:P5) (I4:P5)
    SG->>SG:

    Note over SM,SG,P1,P2,P3,P4,P5,P6: D. Activate Protocol Layers on new processor
    SG->>P1:[ISUP] adsmGoActive: (I3:seqNo=0:mId=0:DisablePeer)(I4:seqNo=0:mId=0:DisablePeer)
    SG->>P2:[ISUP] adsmGoActive: (I3:seqNo=0:mId=0:DisablePeer)(I4:seqNo=0:mId=0:DisablePeer)
    SG->>P3:[ISUP] adsmGoActive: (I3:seqNo=0:mId=0:DisablePeer)(I4:seqNo=0:mId=0:DisablePeer)
    SG->>P4:[ISUP] adsmGoActive: (I3:seqNo=0:mId=0:DisablePeer)(I4:seqNo=0:mId=0:DisablePeer)
    SG->>P5:[ISUP] adsmGoActive: (I3:seqNo=0:mId=0:DisablePeer)(I4:seqNo=0:mId=0:DisablePeer)
    SG->>P6:[ISUP] adsmGoActive: (I3:seqNo=0:mId=0:DisablePeer)(I4:seqNo=0:mId=0:DisablePeer)
    SG->>SG:

    Note over SM,SG,P1,P2,P3,P4,P5,P6: E. For newly created critical Rsets, create shadows on all existing processors
    SG->>SG:

    Note over SM,SG,P1,P2,P3,P4,P5,P6: F. Initiate Neighbor Alive with adjacent Lower Layer
    SG->>P1:[ISUP:I_m] appNeighborAlive: (MTP3)
    P1->>P1:[ISUP:I_m1 to MTP3:M_m] Alive Req
    P1->>P1:[MTP3:M_m to ISUP:I_m1] Alive Cfm
    P1->>SG:[ISUP:I_m to SG] appNeighborAlive Cfm
    P1-->>P1:[MTP3:SM_m] State Update
    SG->>SG:

    Note over SM,SG,P1,P2,P3,P4,P5,P6: G. Initiate Neighbor Alive with adjacent Upper Layer
    SG->>P1:[CC] appNeighborAlive: (ISUP)
    P1->>P1:[CC to ISUP:I_m1] Alive Request
    P1->>P1:[ISUP:I_m1 to CC] Alive Confirm
    P1->>SG:[SG] appNeighborAlive Confirm
    P1-->>P1:[ISUP:S_m1] State Update
    P1-->>P1:[ISUP:S_m1] State Update
    SG->>SM:[SM] scMakeActive: Cfm
  
```

Figure 24

SM	SG	P1	P2	P3	P4	P5	P6
Phase II: processor P ₆ : Make Conventional Protocol Layer MTP2 Active on processor P ₆							
scMakeActive: P ₆ MTP2							
A. Download available User/Provider Rset mapping information							
[RTR] rSetActiveMap: (M _m :P ₃) (M ₁ :P ₃) (M ₂ :P ₄) (M ₃ :P ₄)							
B. Create existing critical resource Shadows on new processor if don't exist							
C. Download new mappings to adjacent routers							
D. Activate Protocol Layers on new processor							
NOTE: Since MTP2 is a conventional protocol layer, it goes into the active state soon after it has been configured. The System Manager does not send a adsmGoActive request for this type of protocol layers.							
E. For newly created critical Rsets, create shadows on all existing processors							
F. Initiate Neighbor Alive with adjacent Lower Layer							

Figure 25

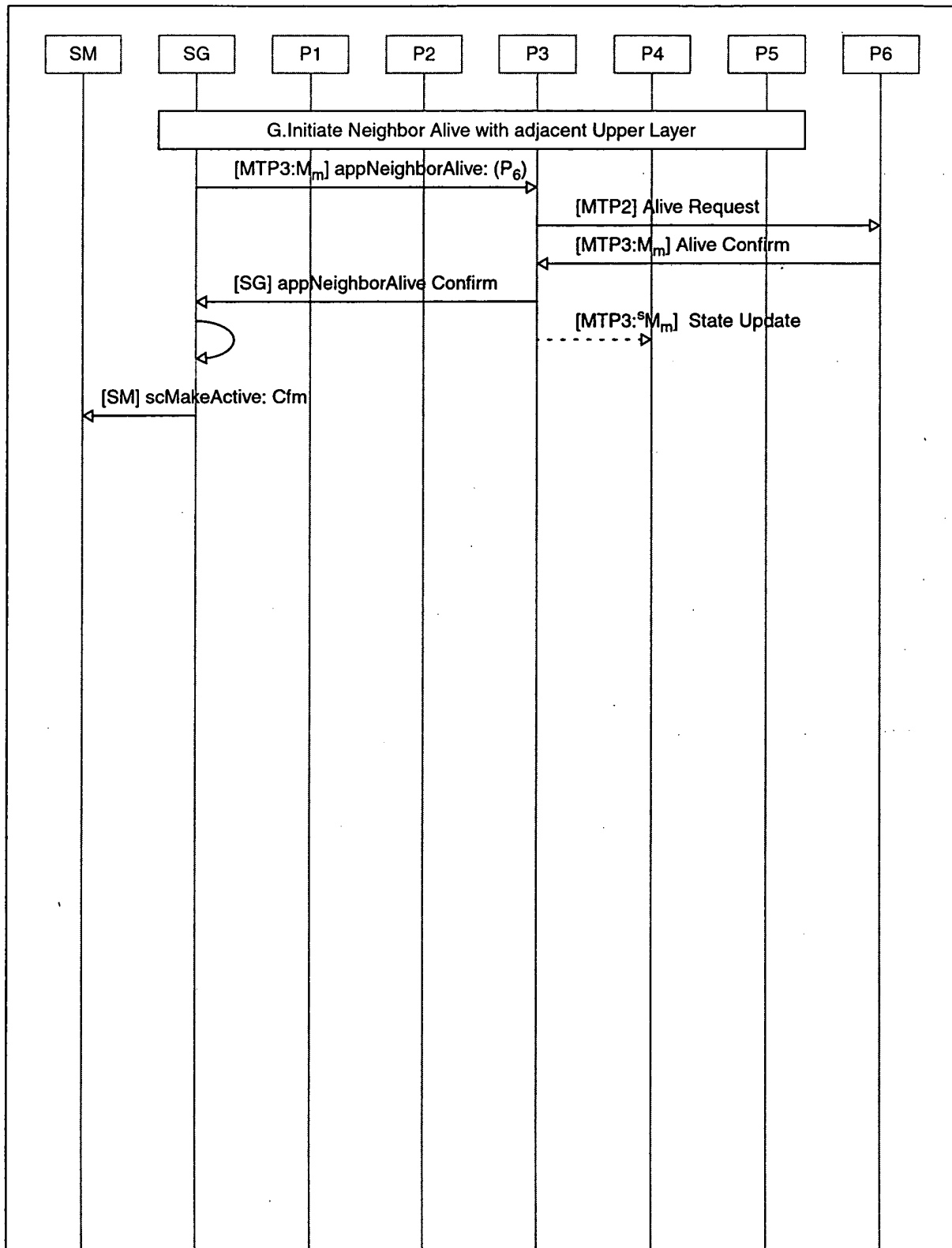


Figure 26

Phase I

1. Make M_1 , M_2 and M_3 of MTP3 standby on P_5

Phase II

2. Make CC standby on P_2

Legend:

- Out of Service Element (hatched box)
- Active Element (solid box)
- Standby Element (vertically striped box)

shadow resource sets not shown

Figure 27

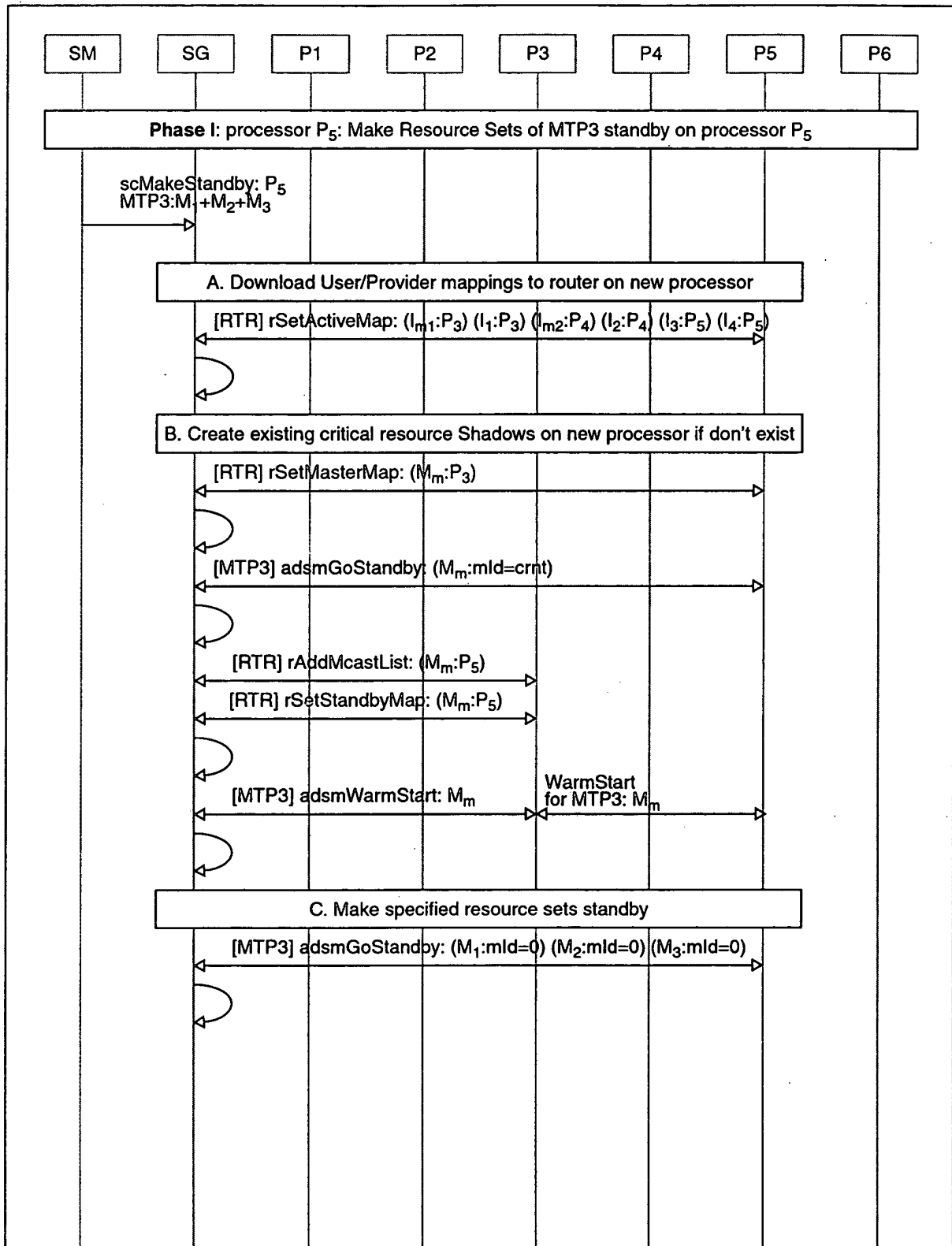


Figure 28

```

sequenceDiagram
    participant SM
    participant SG
    participant P1
    participant P2
    participant P3
    participant P4
    participant P5
    participant P6

    Note over SG, P1, P2, P3, P4, P5: D. Update Router(s) on active processor(s) about new standby mappings
    SG->>P1: [RTR] rSetStandbyMap: (M1:P5)
    SG->>P2: [RTR] rSetStandbyMap: (M2:P5) (M3:P5)
    SG->>P3: 
    SG->>P4: 
    SG->>P5: 
    SG->>P6: 

    Note over SG, P1, P2, P3, P4, P5: E. Make Actives WarmStart the new Standbys
    SG->>P1: [MTP3] adsmWarmStart: (M1)
    Note over P1: WarmStart for MTP3: M1
    SG->>P2: [MTP3] adsmWarmStart: (M2) (M3)
    Note over P2: WarmStart for MTP3: M2
    Note over P3: WarmStart for MTP3: M3
    SG->>P4: 
    SG->>P5: 
    SG->>P6: 

    SG->>SM: [SM]scMakeStandby: Cfm
  
```

Figure 29

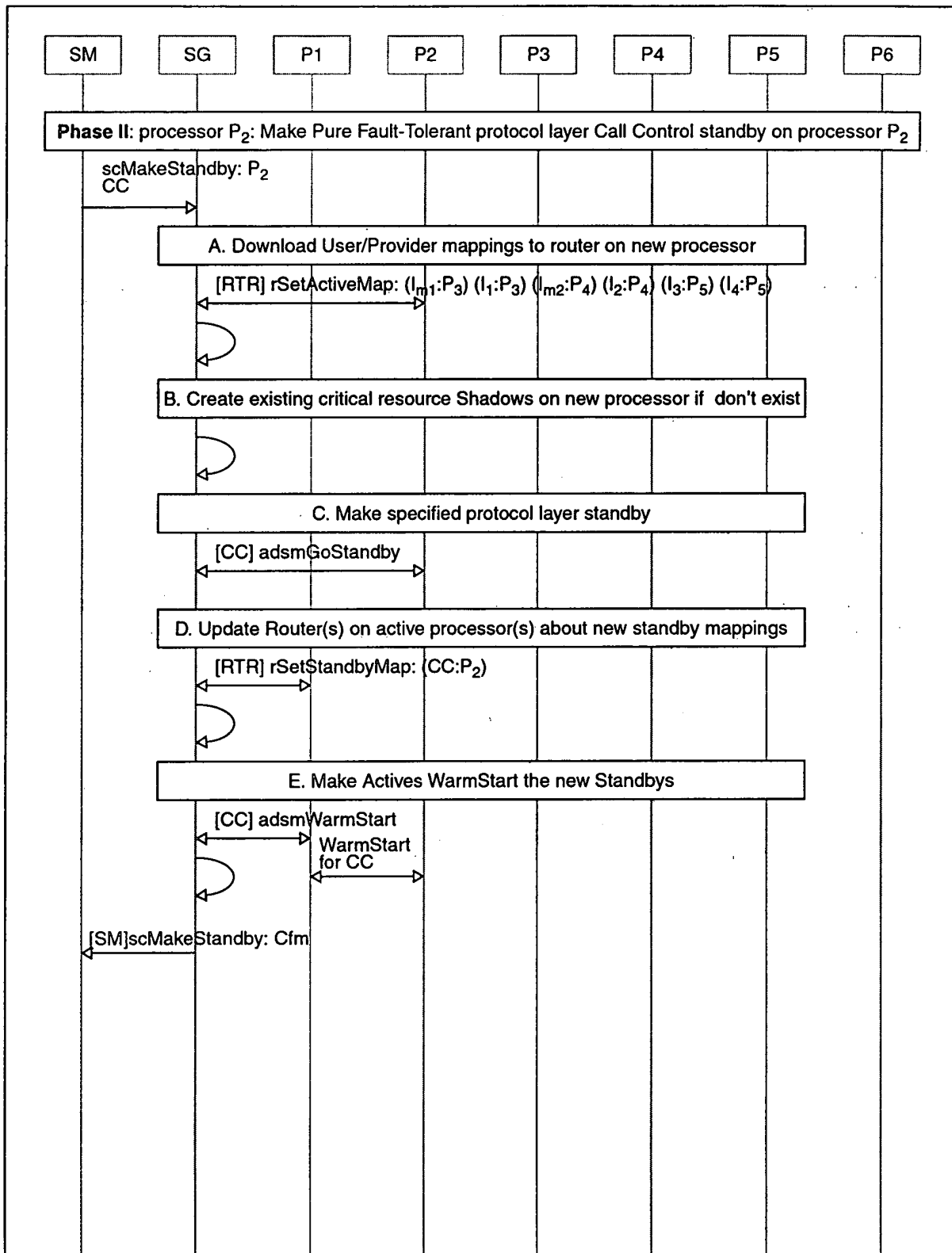


Figure 30

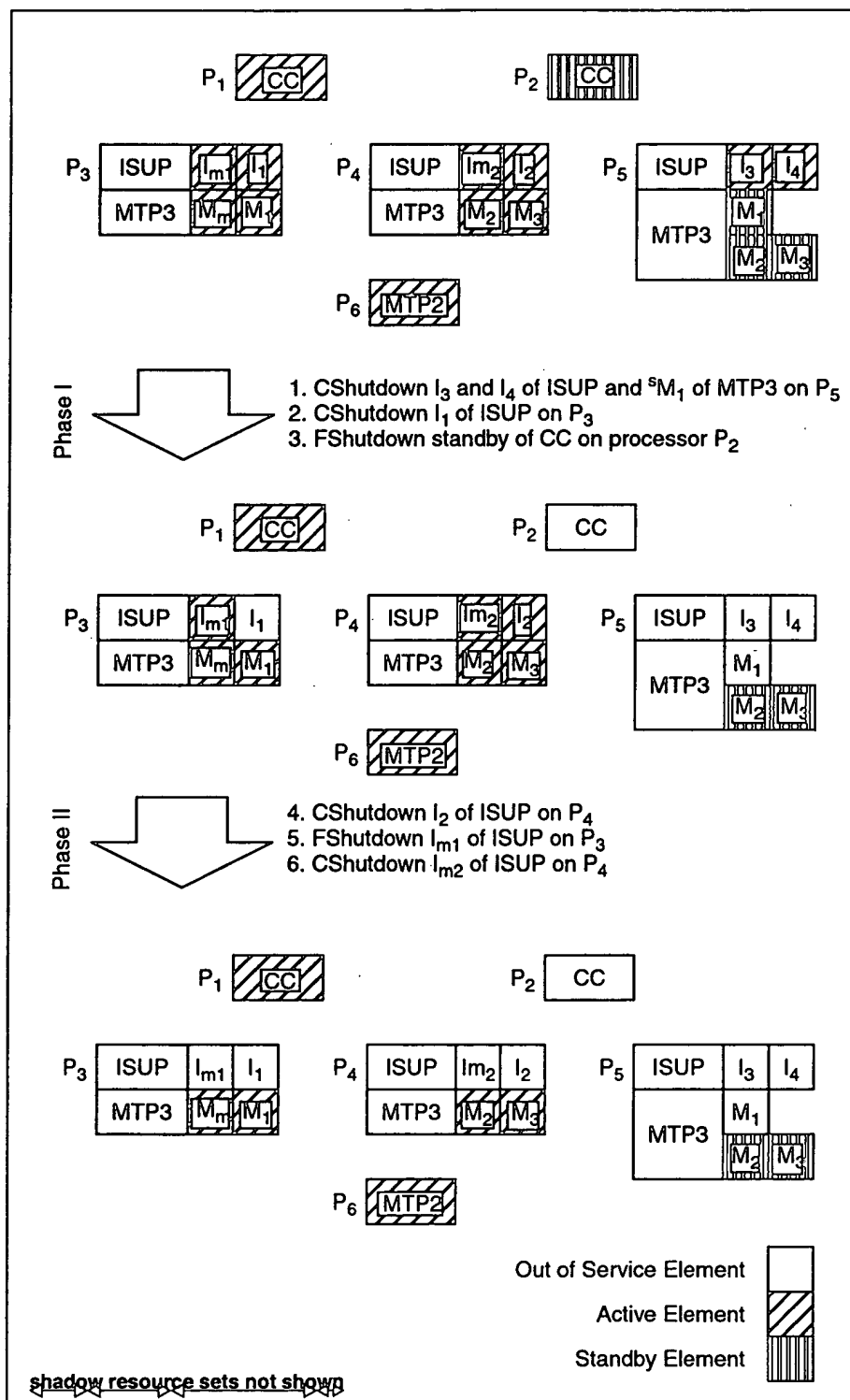


Figure 31

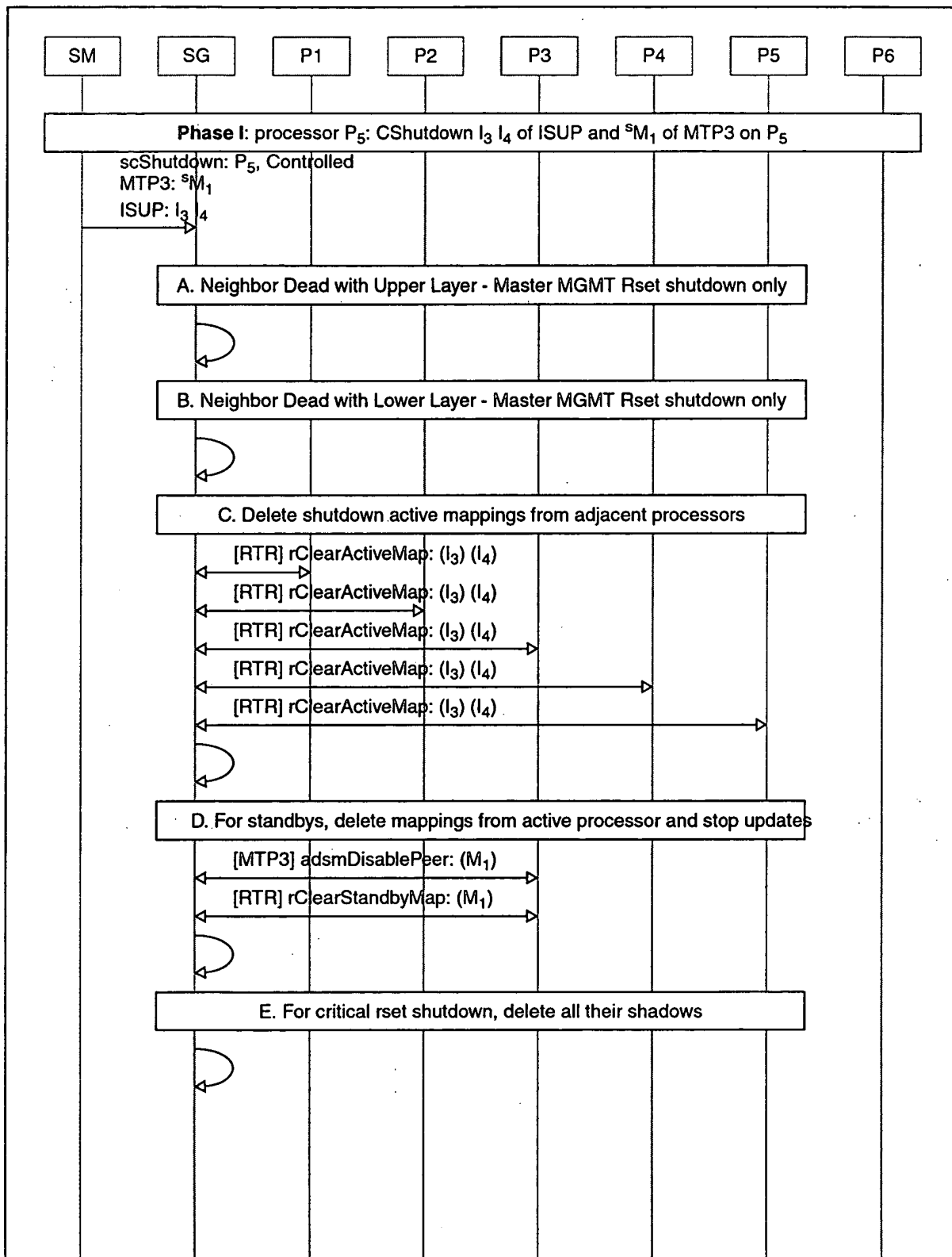


Figure 32

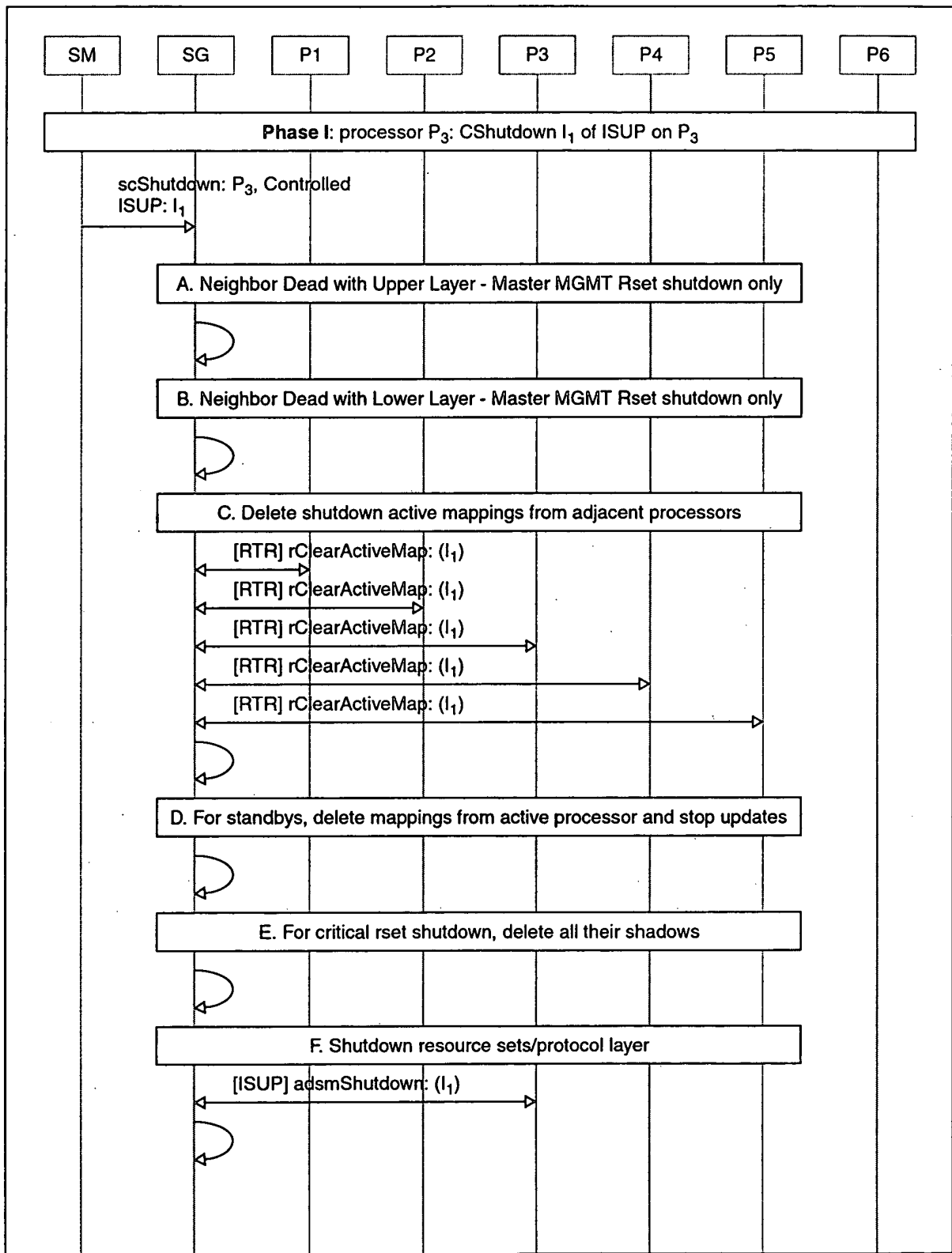


Figure 34

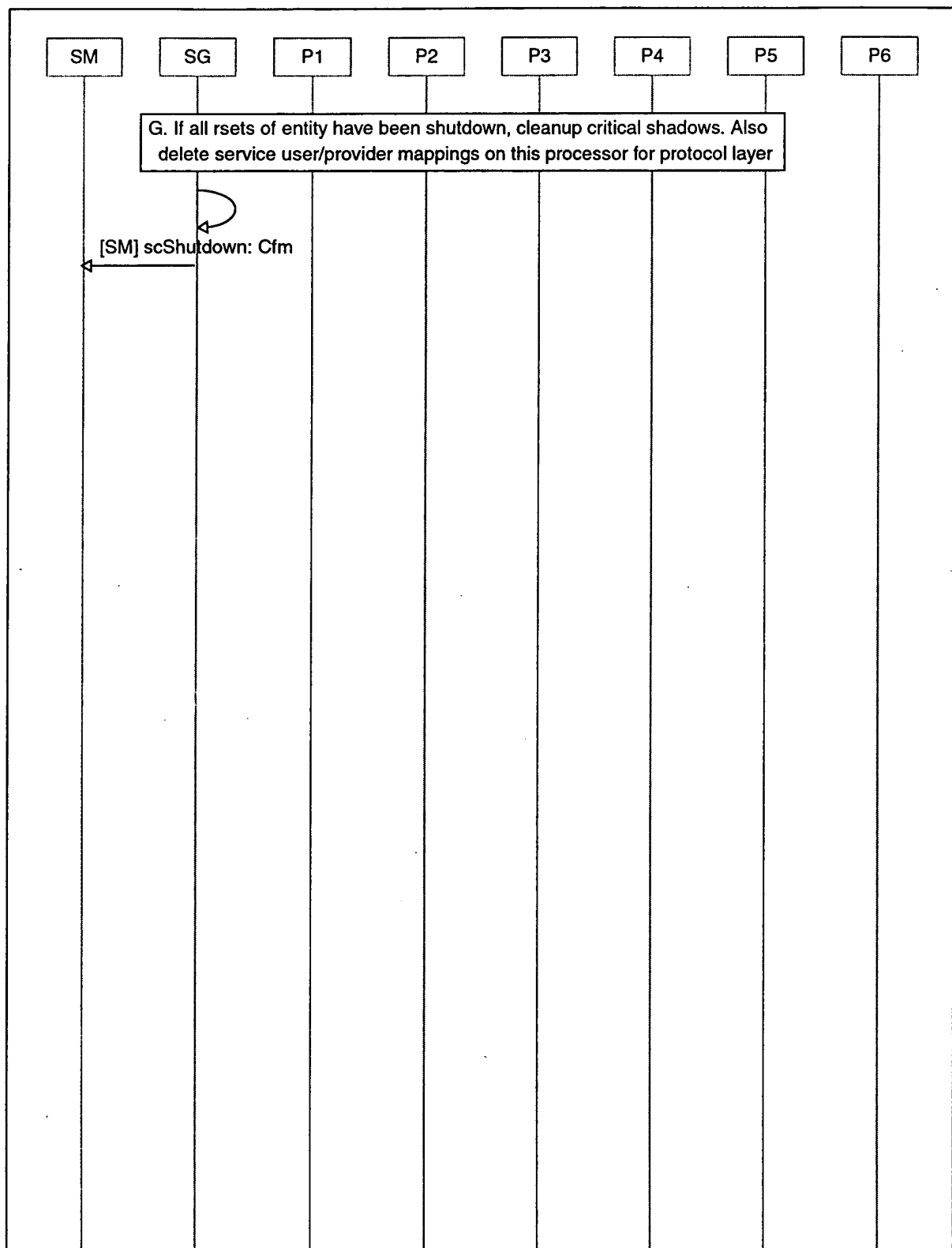


Figure 35

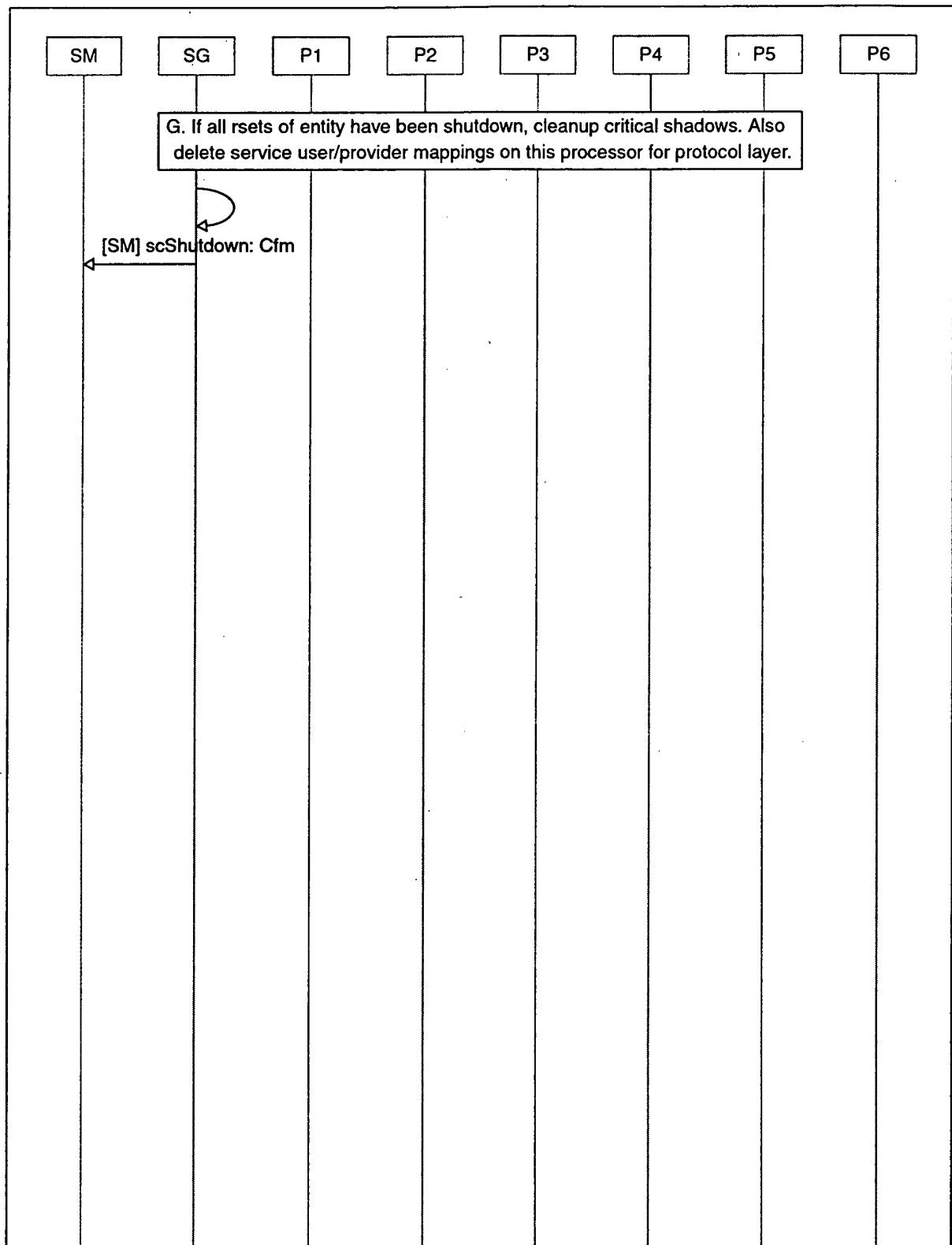


Figure 37

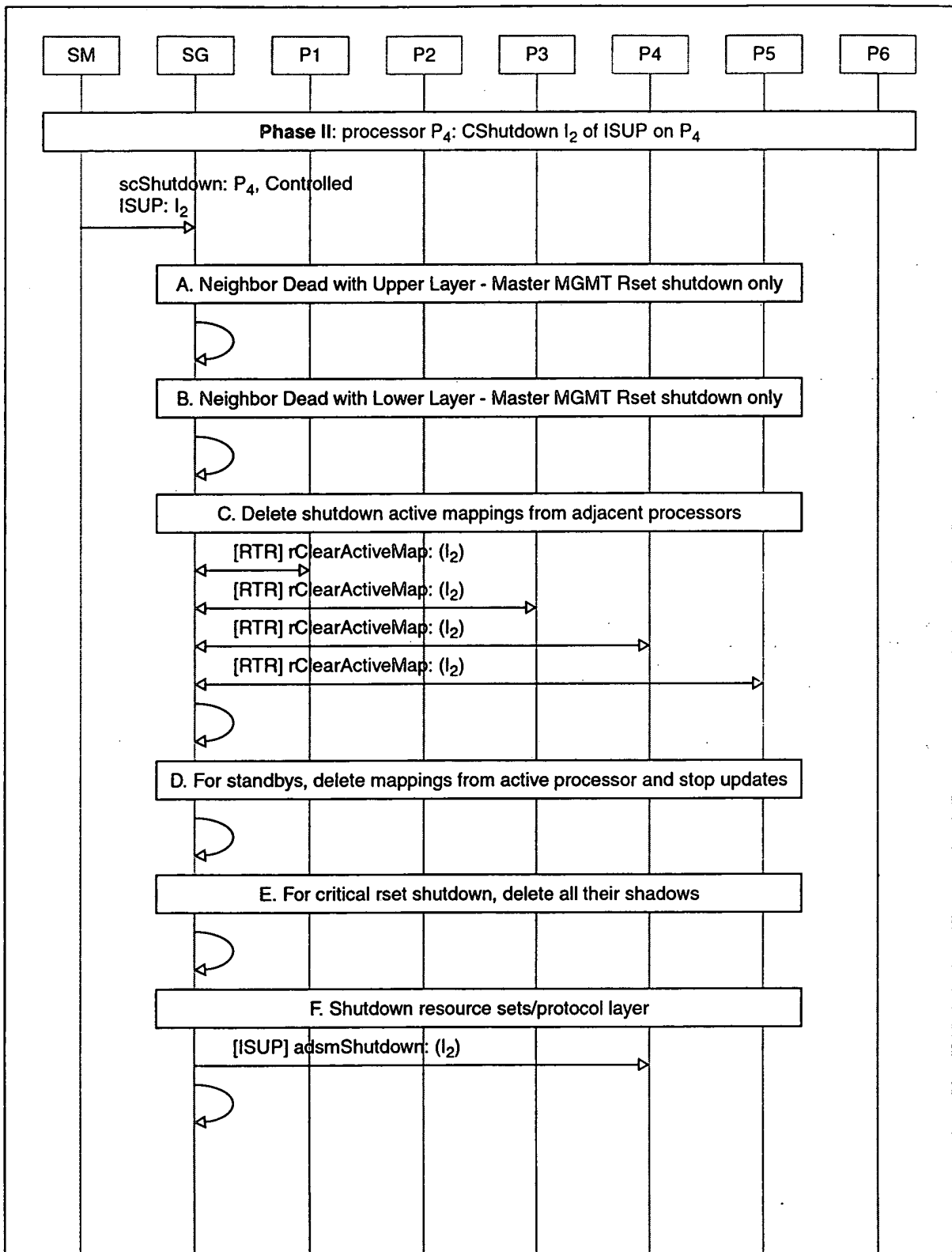


Figure 38

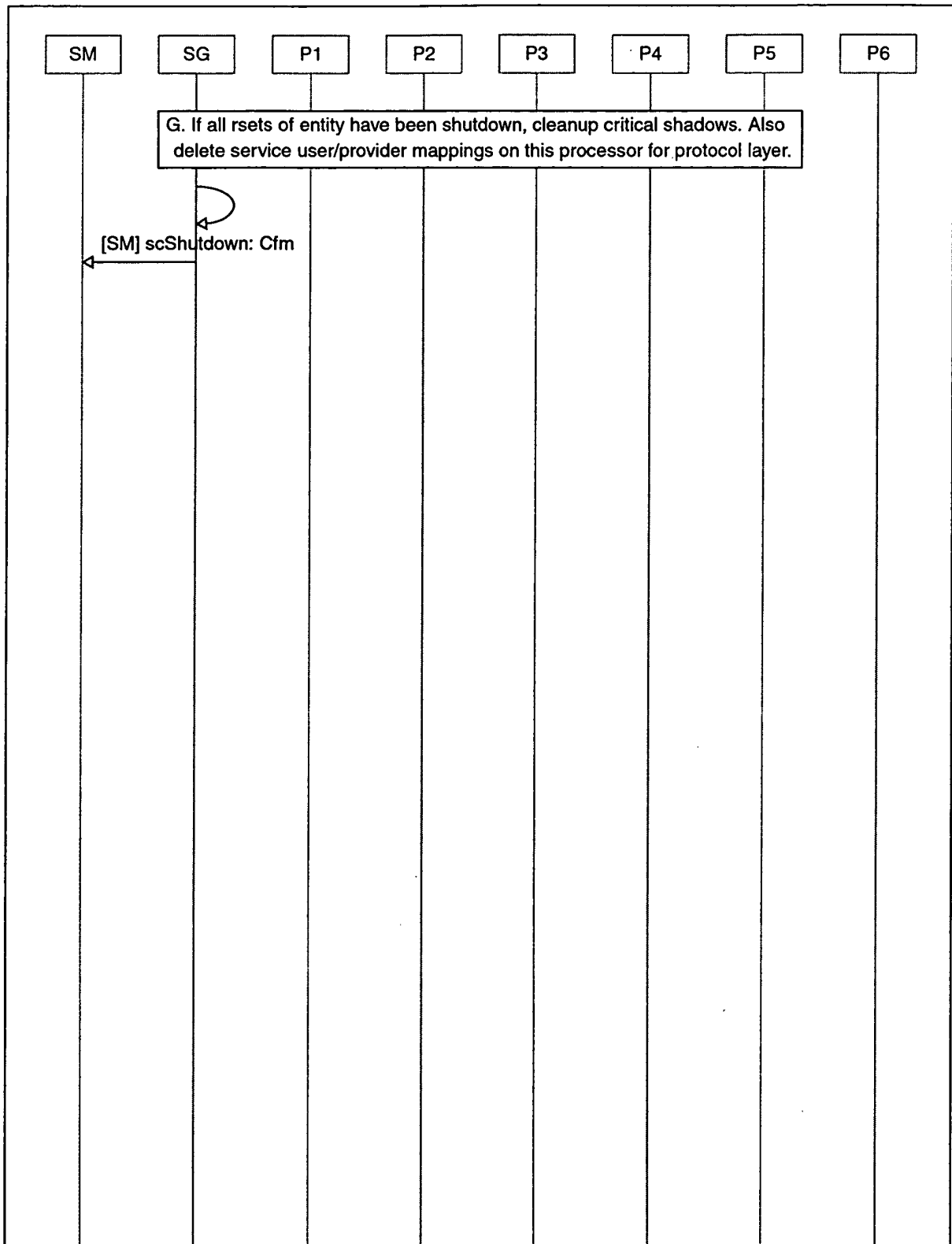


Figure 39

```

sequenceDiagram
    participant SM
    participant SG
    participant P1
    participant P2
    participant P3
    participant P4
    participant P5
    participant P6

    Note over SM, SG, P1, P2, P3, P4, P5, P6: Phase II: processor P3: FShutdown I_m1 of ISUP on P3

    SM->>SG: scShutdown: P3, Forced  
ISUP: I_m1

    Note over SM, SG, P1, P2, P3, P4, P5, P6: A. Neighbor Dead with Upper Layer - Master MGMT Rset shutdown only

    SG->>P1: [CC] appNeighborDead: (ISUP)
    P1-->>SG: [ISUP:I_m1] appNeighborDead: (CC)
    P1-->>P1: 
    P1-->>P2: 
    P1-->>P3: 
    P1-->>P4: 
    P1-->>P5: 
    P1-->>P6: 

    Note over SM, SG, P1, P2, P3, P4, P5, P6: B. Neighbor Dead with Lower Layer - Master MGMT Rset shutdown only

    P1-->>P3: [ISUP:I_m1] appNeighborDead: (MTP3)
    P3-->>P1: [MTP3:M_m] appNeighborDead: (ISUP)
    P1-->>P2: 
    P1-->>P3: 
    P1-->>P4: 
    P1-->>P5: 
    P1-->>P6: 

    Note over SM, SG, P1, P2, P3, P4, P5, P6: C. Delete shutdown active mappings from adjacent processors

    P1-->>P2: [RTR] rClearActiveMap: (I_m1)
    P1-->>P3: [RTR] rClearActiveMap: (I_m1)
    P1-->>P4: [RTR] rClearActiveMap: (I_m1)
    P1-->>P5: [RTR] rClearActiveMap: (I_m1)
    P1-->>P6: [RTR] rClearActiveMap: (I_m1)
    P1-->>P1: 

    Note over SM, SG, P1, P2, P3, P4, P5, P6: D. For standbys, delete mappings from active processor and stop updates

    P1-->>P1: 
  
```

Figure 40

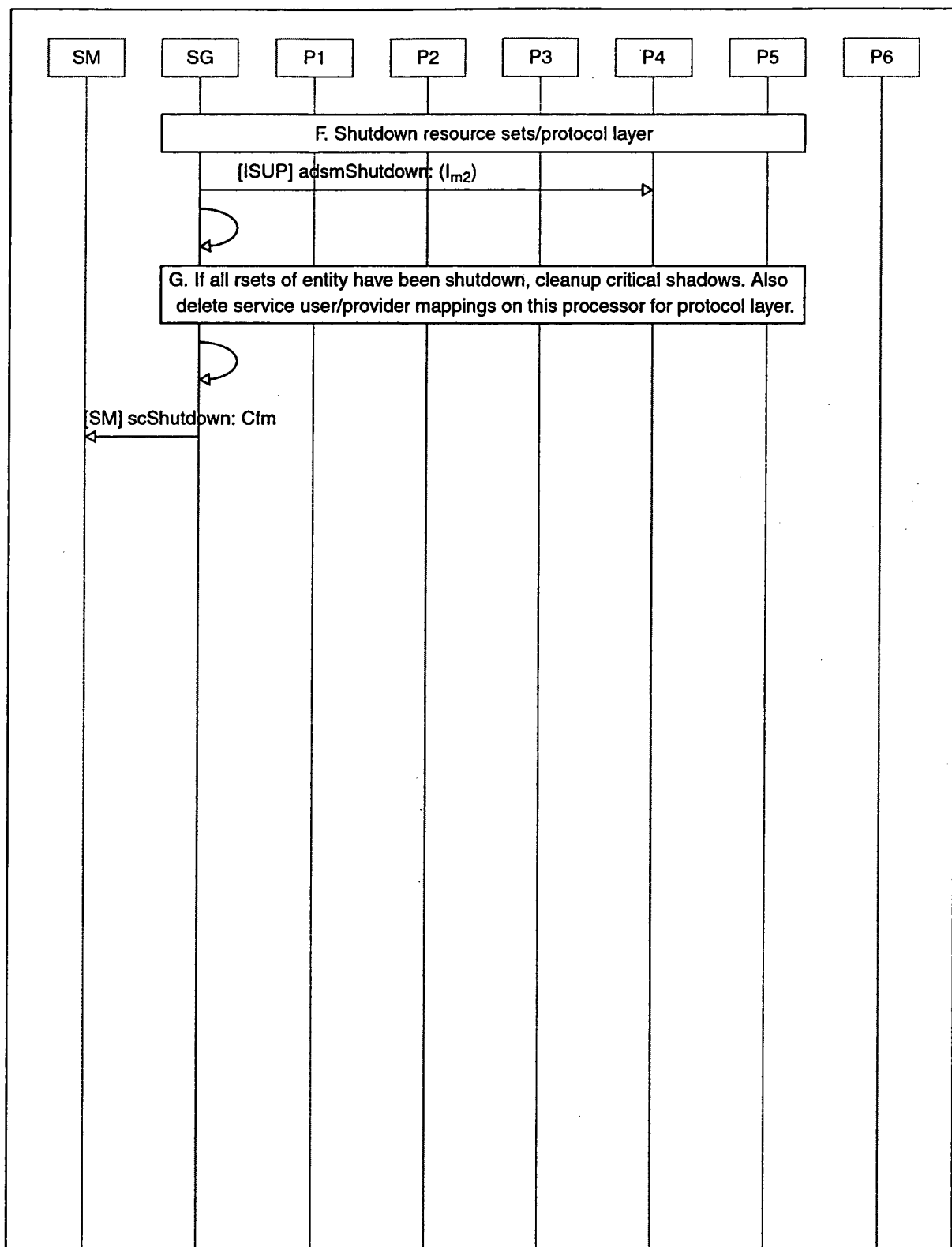


Figure 43

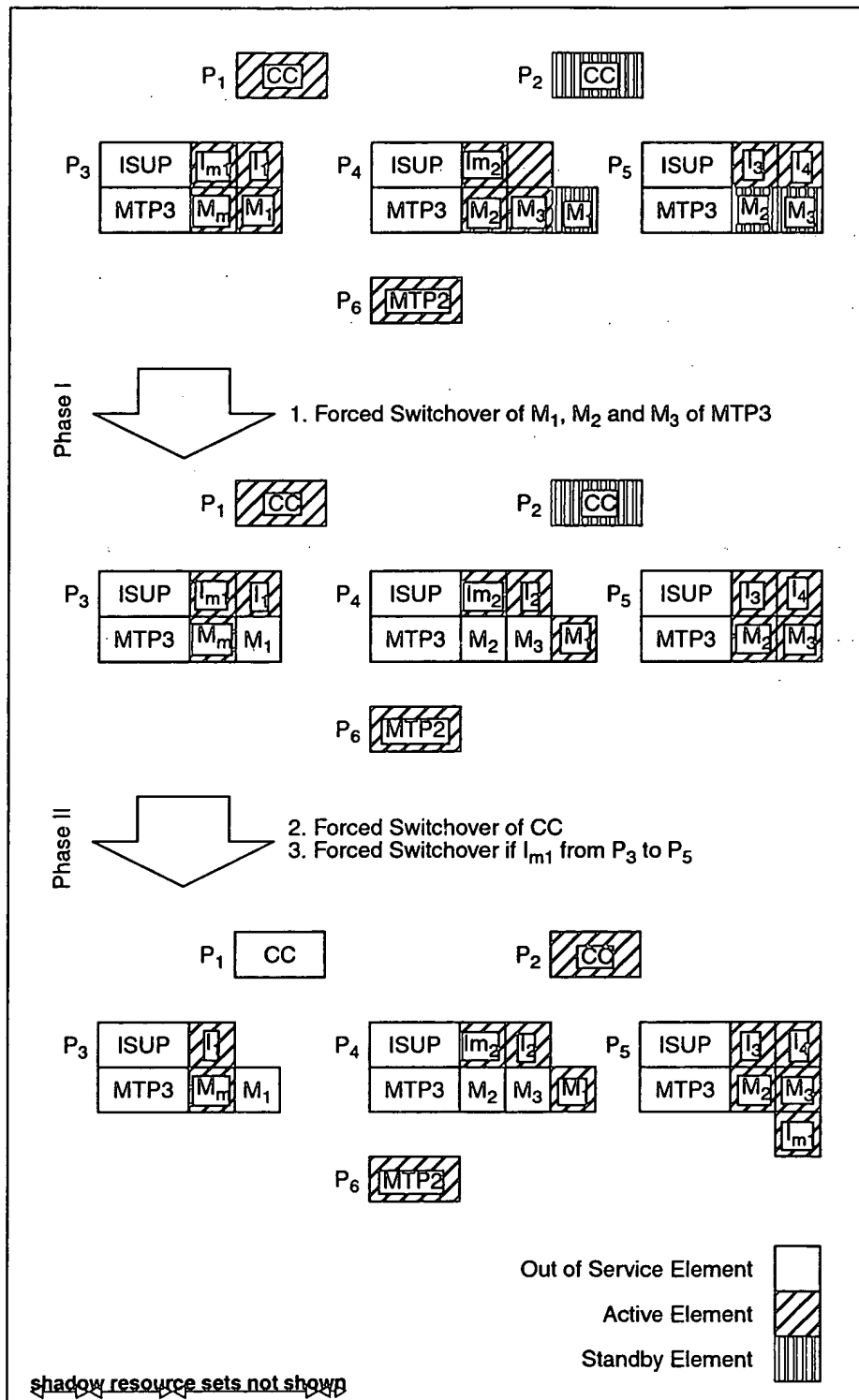


Figure 44

```

sequenceDiagram
    participant SM
    participant SG
    participant P1
    participant P2
    participant P3
    participant P4
    participant P5
    participant P6

    Note over SM, P6: Phase I: Forced Switchover of M1, M2 and M3 of MTP3
    SM->>SG: scForcedSwitchover:  
MTP3: [(M1) (M2) (M3) []]

    Note over SM, P6: A. Hold messages at adjacent upper and lower layers
    SM->>P1: [RTR] rHoldQueue: (M1) (M2) (M3)
    P1-->>P2: [RTR] rHoldQueue: (M1) (M2) (M3)
    P2-->>P3: [RTR] rHoldQueue: (M1) (M2) (M3)
    P3-->>P4: [RTR] rHoldQueue: (M1) (M2) (M3)
    P4-->>P5: [RTR] rHoldQueue: (M1) (M2) (M3)
    P5-->>P6: [RTR] rHoldQueue: (M1) (M2) (M3)
    P6-->>SM: [RTR] rHoldQueue: (M1) (M2) (M3)
    SM->>SM: self-loop

    Note over SM, P6: B. Delete standby mapping on old active processor
    SM->>P1: [RTR] rClearStandbyMap: (M1)
    P1-->>P2: [RTR] rClearStandbyMap: (M2 M3)
    P2-->>P3: [RTR] rClearStandbyMap: (M2 M3)
    P3-->>P4: [RTR] rClearStandbyMap: (M2 M3)
    P4-->>P5: [RTR] rClearStandbyMap: (M2 M3)
    P5-->>P6: [RTR] rClearStandbyMap: (M2 M3)
    P6-->>SM: [RTR] rClearStandbyMap: (M2 M3)
    SM->>SM: self-loop

    Note over SM, P6: C. Delete active mapping on new active processor
    SM->>P1: [RTR] rClearActiveMap: (M1)
    P1-->>P2: [RTR] rClearActiveMap: (M2) (M3)
    P2-->>P3: [RTR] rClearActiveMap: (M2) (M3)
    P3-->>P4: [RTR] rClearActiveMap: (M2) (M3)
    P4-->>P5: [RTR] rClearActiveMap: (M2) (M3)
    P5-->>P6: [RTR] rClearActiveMap: (M2) (M3)
    P6-->>SM: [RTR] rClearActiveMap: (M2) (M3)
    SM->>SM: self-loop

    Note over SM, P6: D. Download new mappings to adjacent routers
    SM->>P1: [RTR] rSetActiveMap: (M1:P4) (M2:P5) (M3:P5)
    P1-->>P2: [RTR] rSetActiveMap: (M1:P4) (M2:P5) (M3:P5)
    P2-->>P3: [RTR] rSetActiveMap: (M1:P4) (M2:P5) (M3:P5)
    P3-->>P4: [RTR] rSetActiveMap: (M1:P4) (M2:P5) (M3:P5)
    P4-->>P5: [RTR] rSetActiveMap: (M1:P4) (M2:P5) (M3:P5)
    P5-->>P6: [RTR] rSetActiveMap: (M1:P4) (M2:P5) (M3:P5)
    P6-->>SM: [RTR] rSetActiveMap: (M1:P4) (M2:P5) (M3:P5)
    SM->>SM: self-loop
    
```

Figure 45

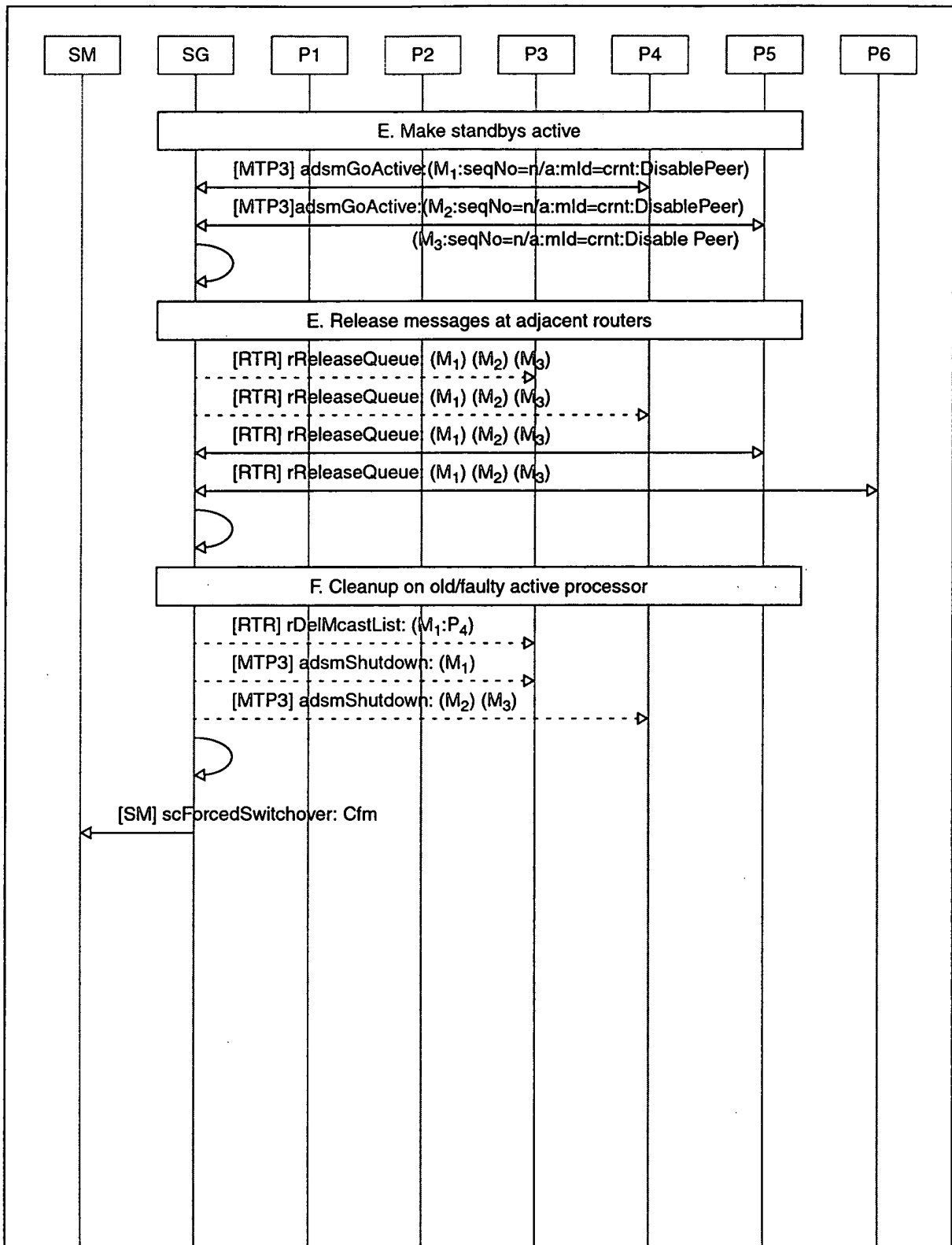


Figure 46

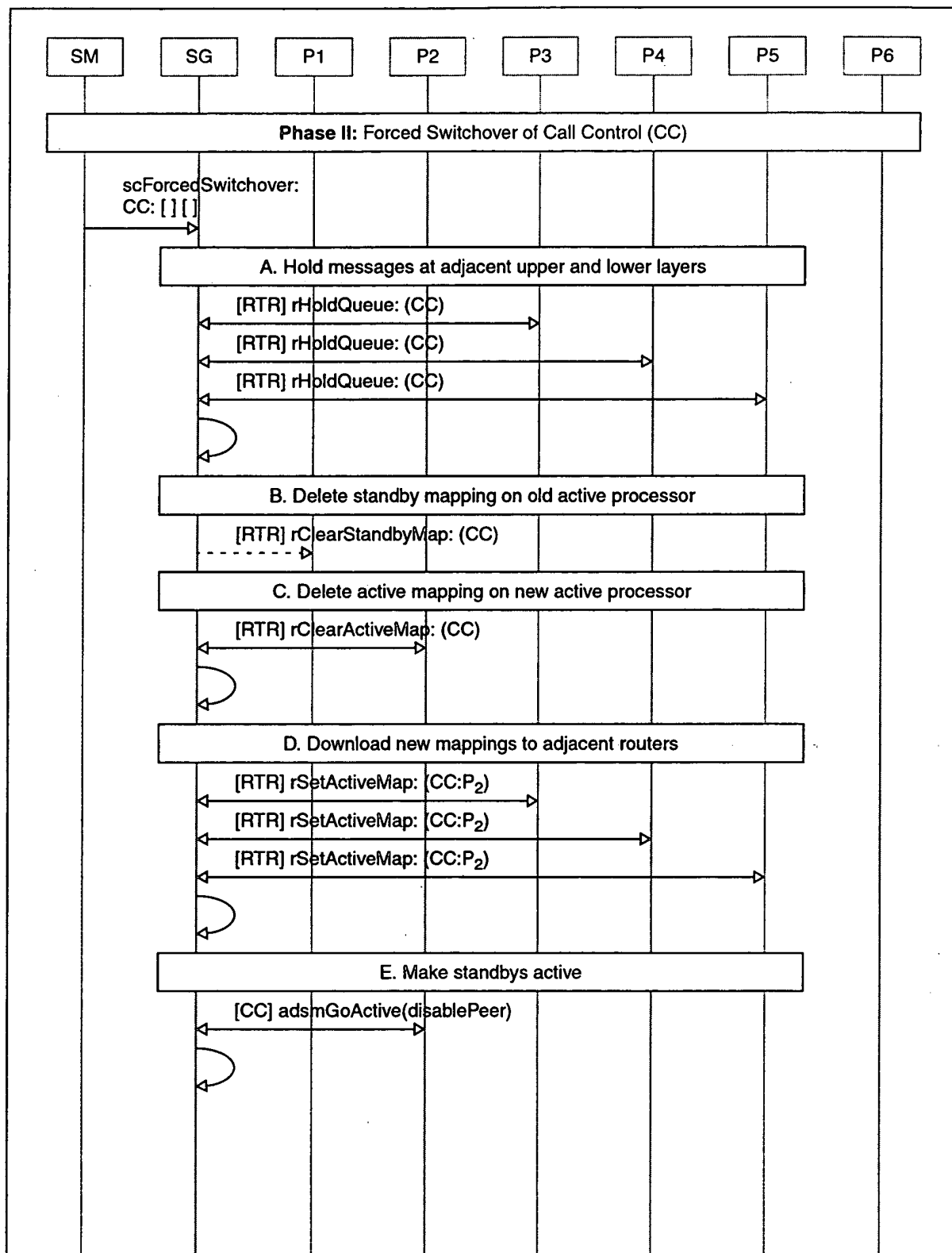


Figure 47

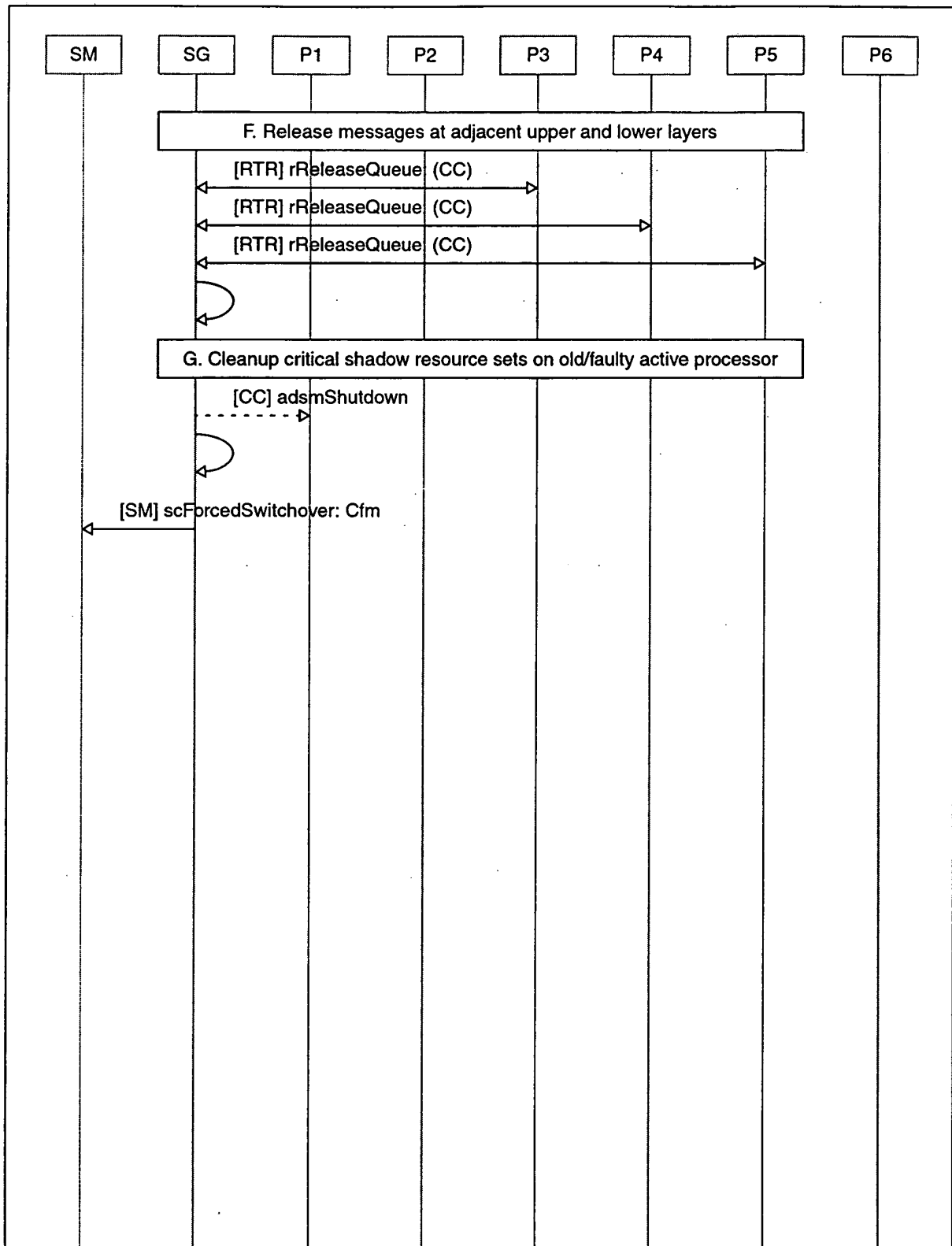


Figure 48

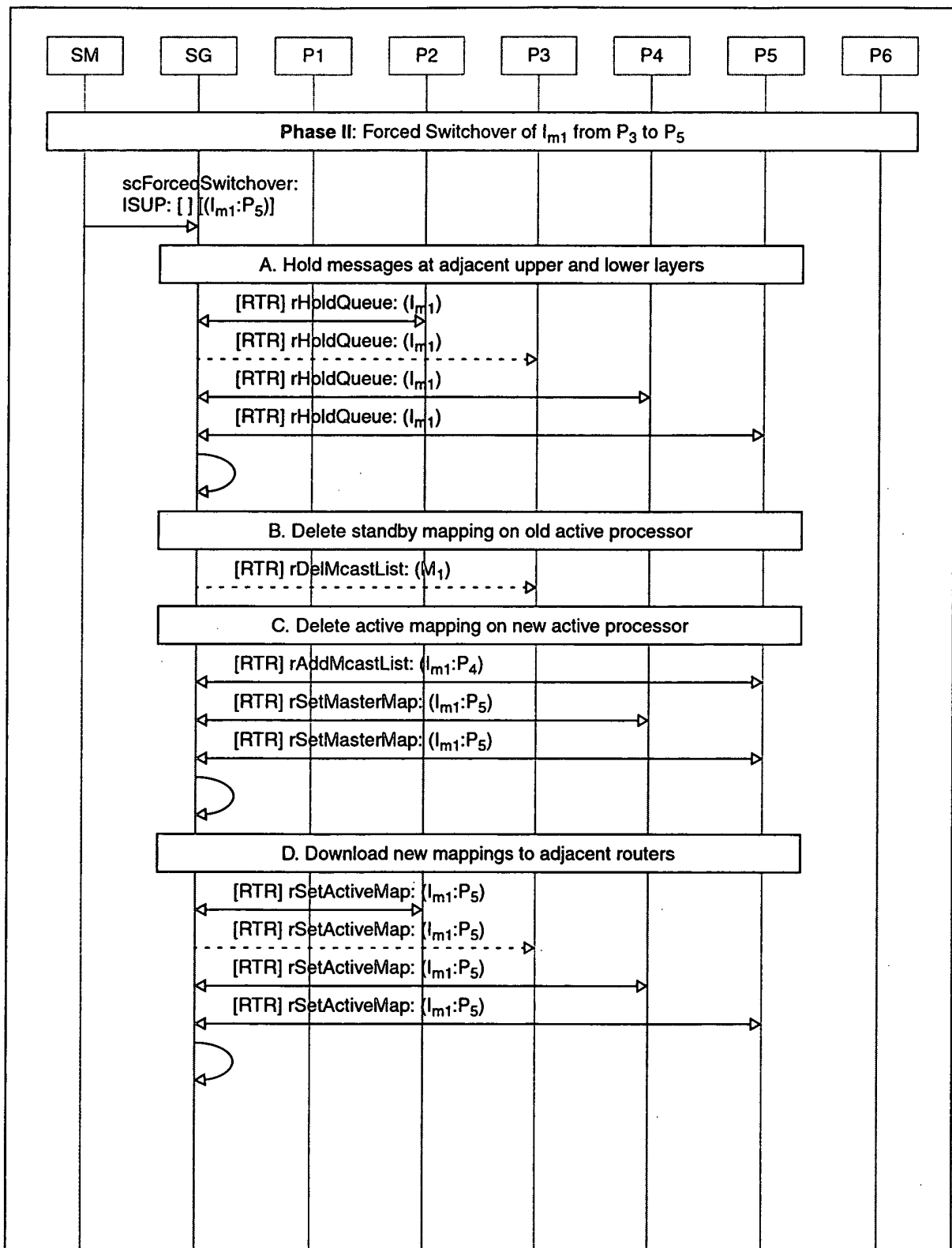


Figure 49

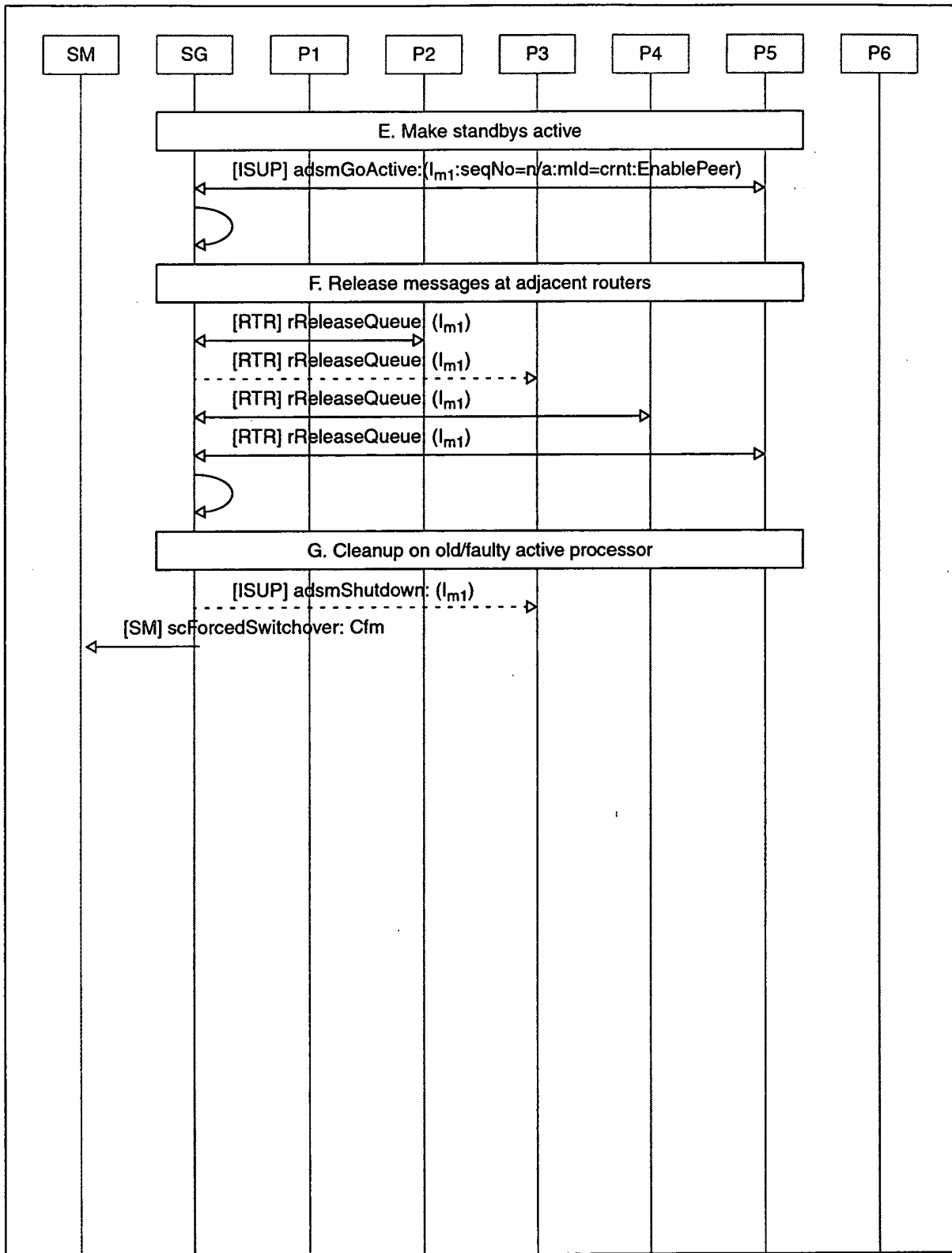


Figure 50

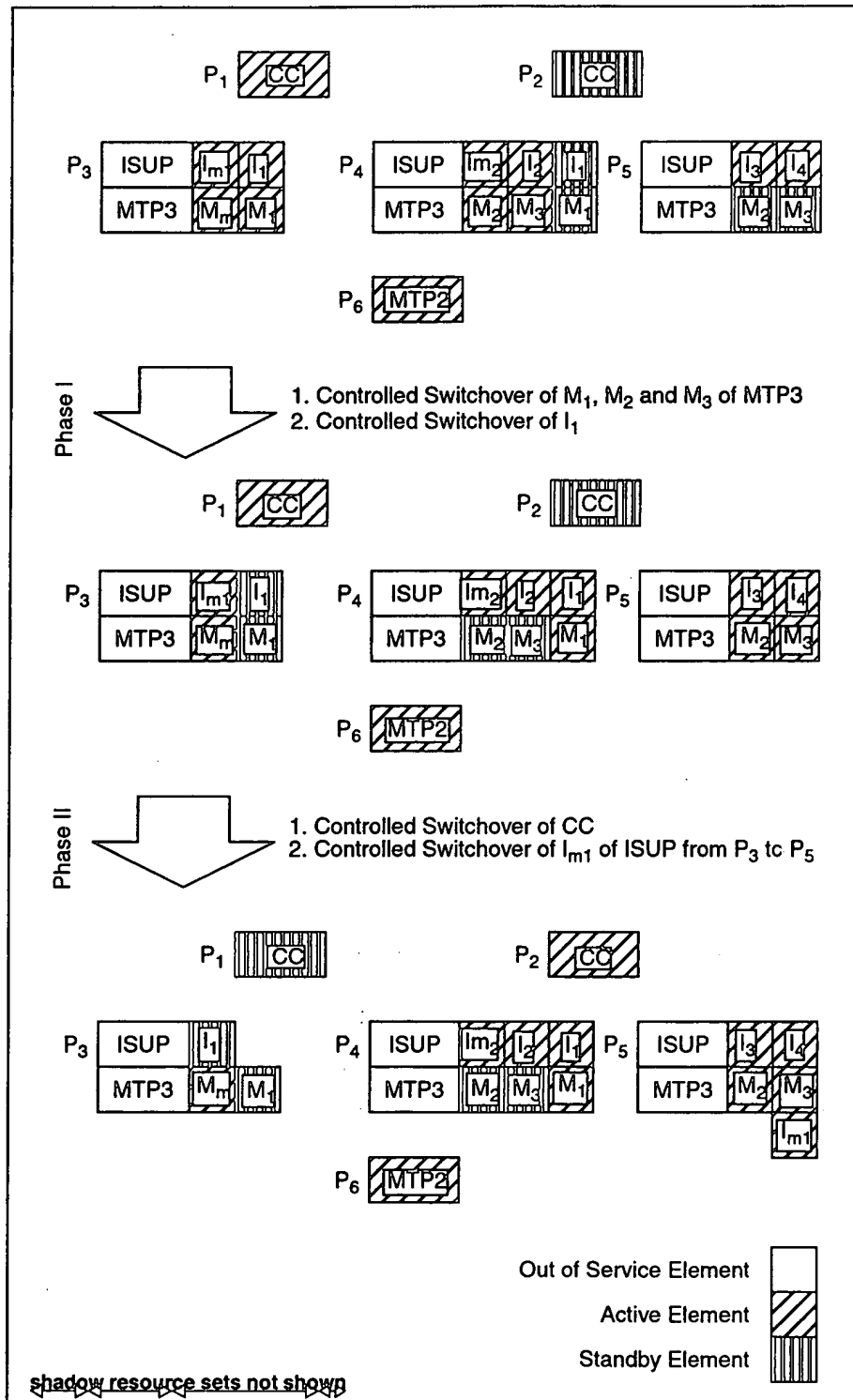


Figure 51

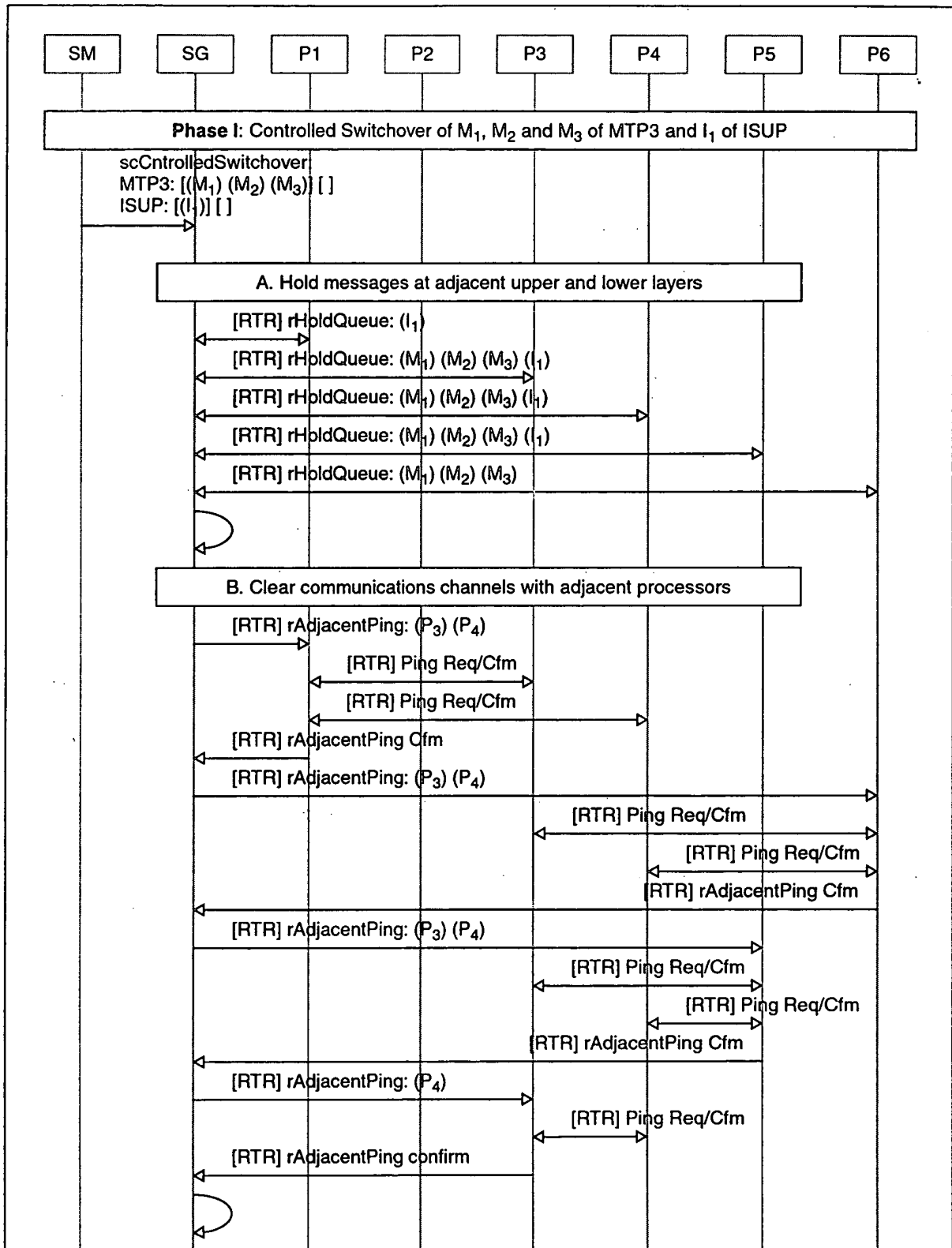


Figure 52

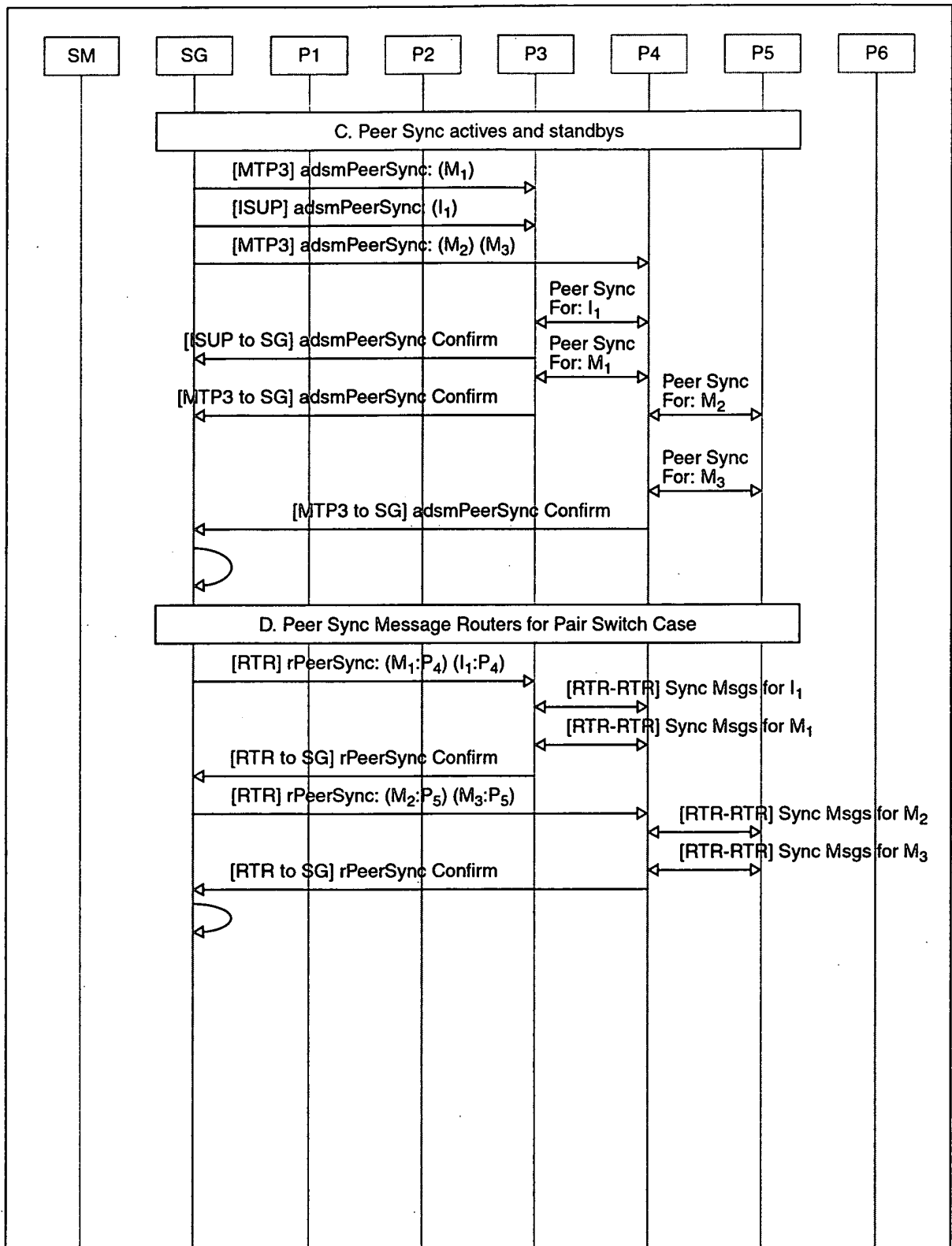


Figure 53

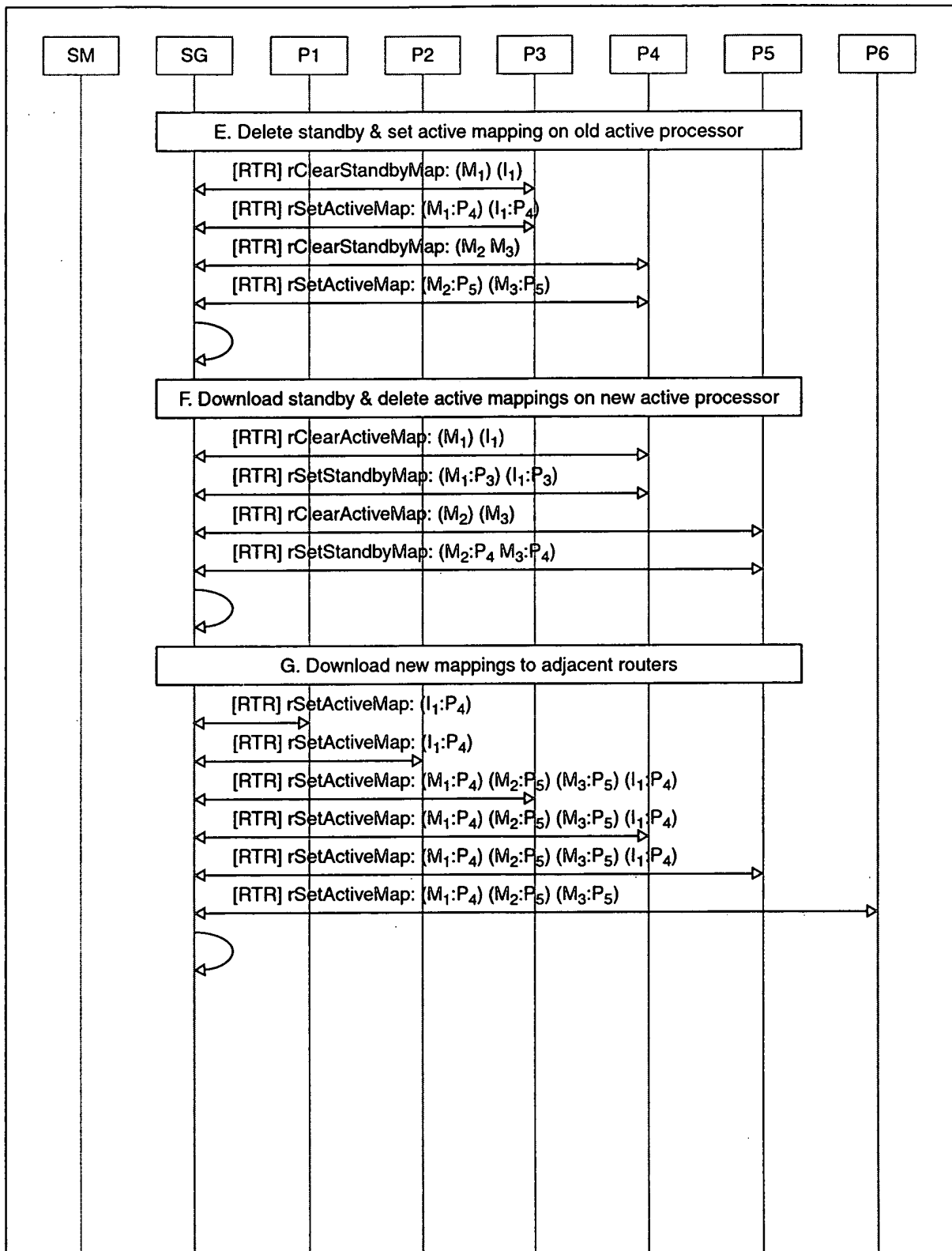
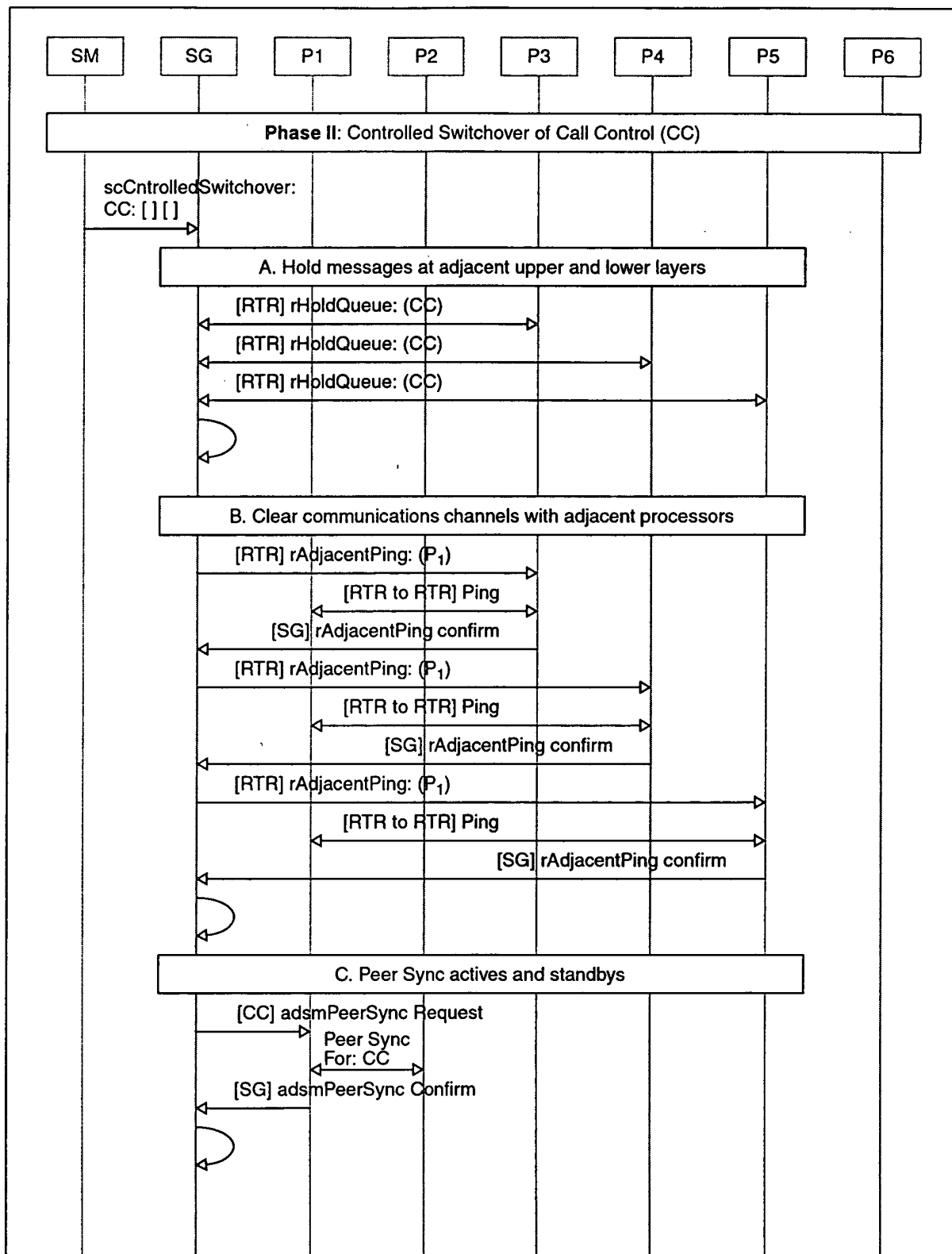


Figure 54



```

sequenceDiagram
    participant SM
    participant SG
    participant P1
    participant P2
    participant P3
    participant P4
    participant P5
    participant P6

    Note over SG: D. Peer Sync Message Routers for Pair Switch Case
    SG->>SG: 
    Note over SG: E. Delete standby & set active mappings on old active processor
    SG->>P1: [RTR] rClearStandbyMap: (CC)
    P1->>SG: 
    SG->>P2: [RTR] rSetActiveMap: (CC:P2)
    P2->>SG: 
    SG->>SG: 
    Note over SG: F. Download standby & delete active mappings on new active processor
    SG->>P2: [RTR] rClearActiveMap: (CC)
    P2->>SG: 
    SG->>P1: [RTR] rSetStandbyMap: (CC:P1)
    P1->>SG: 
    SG->>SG: 
    Note over SG: G. Download new mappings to adjacent routers
    SG->>P1: [RTR] rSetActiveMap: (CC:P2)
    P1->>SG: 
    SG->>P2: [RTR] rSetActiveMap: (CC:P2)
    P2->>SG: 
    SG->>P3: [RTR] rSetActiveMap: (CC:P2)
    P3->>SG: 
    SG->>P4: [RTR] rSetActiveMap: (CC:P2)
    P4->>SG: 
    SG->>P5: [RTR] rSetActiveMap: (CC:P2)
    P5->>SG: 
    SG->>SG: 
    Note over SG: H. Make actives standby
    SG->>P1: [CC] adsmGoStandby
    P1->>SG: 
    SG->>SG: 
    Note over SG: I. Make standbys active
    SG->>P1: [CC] adsmGoActive(EnablePeer)
    P1->>SG: 
    SG->>SG: 
  
```

Figure 57

```

sequenceDiagram
    participant SM
    participant SG
    participant P1
    participant P2
    participant P3
    participant P4
    participant P5
    participant P6

    Note over SG: J. Release messages for Peer Switch Case
    SG->>SG: 
    Note over SG: K. Release messages at adjacent processors
    SG->>P3: [RTR] rReleaseQueue (CC)
    SG->>P4: [RTR] rReleaseQueue (CC)
    SG->>P5: [RTR] rReleaseQueue (CC)
    SG->>SG: 
    SG->>SM: [SM] scControlledSwitchover: Cfm
  
```

Figure 58

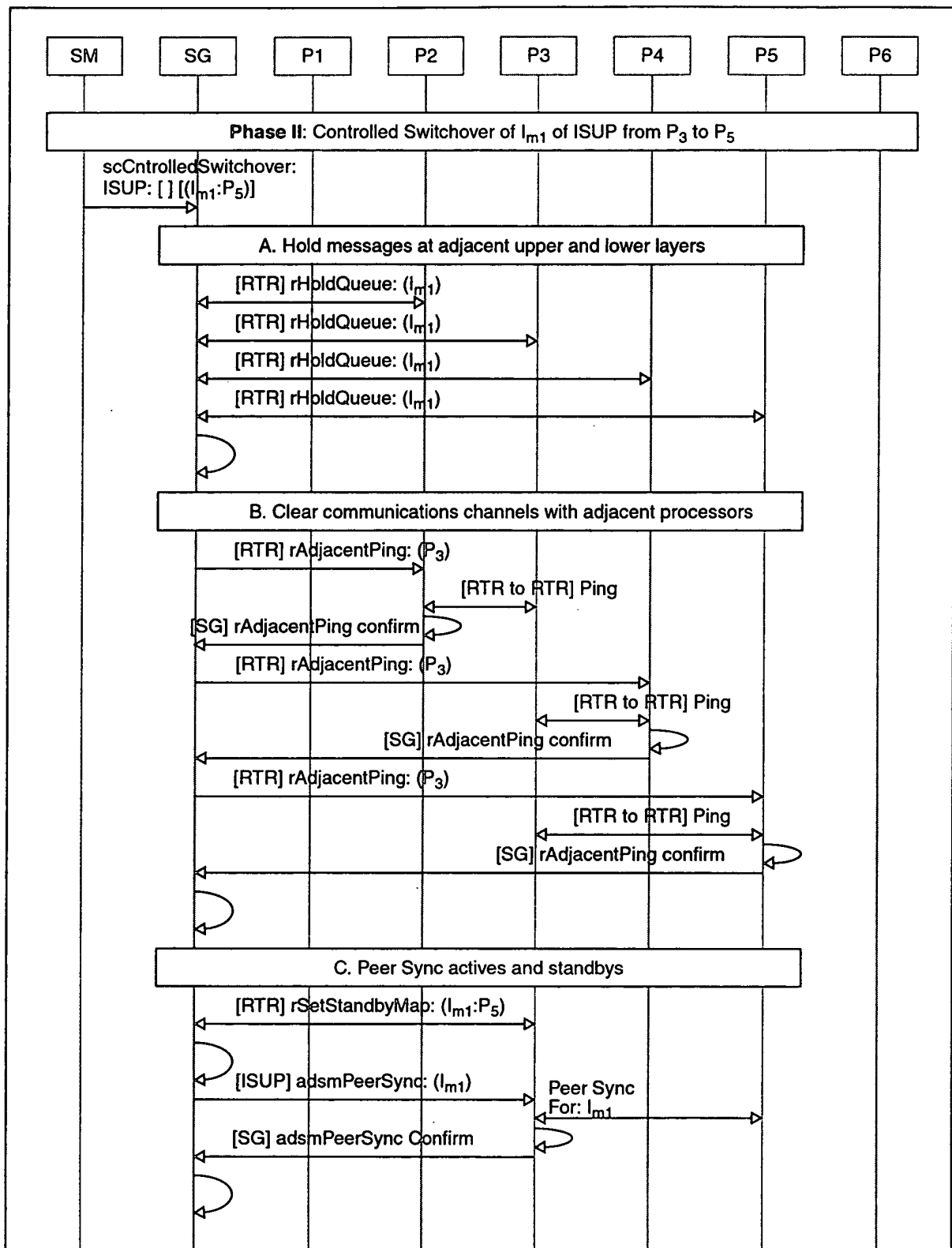


Figure 59

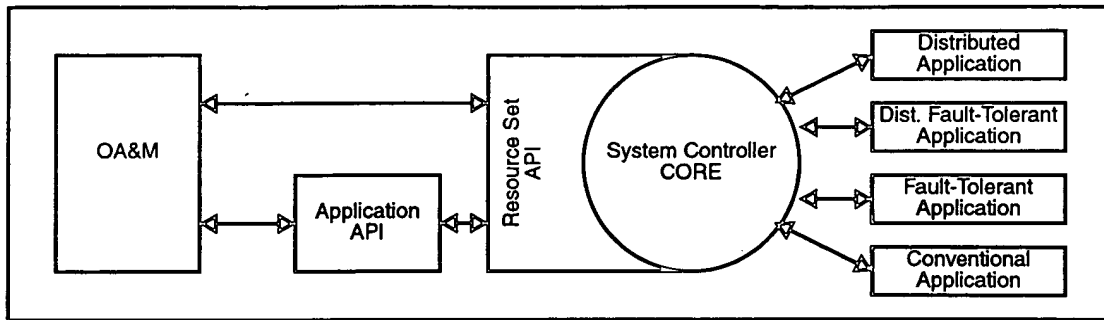


Figure 62

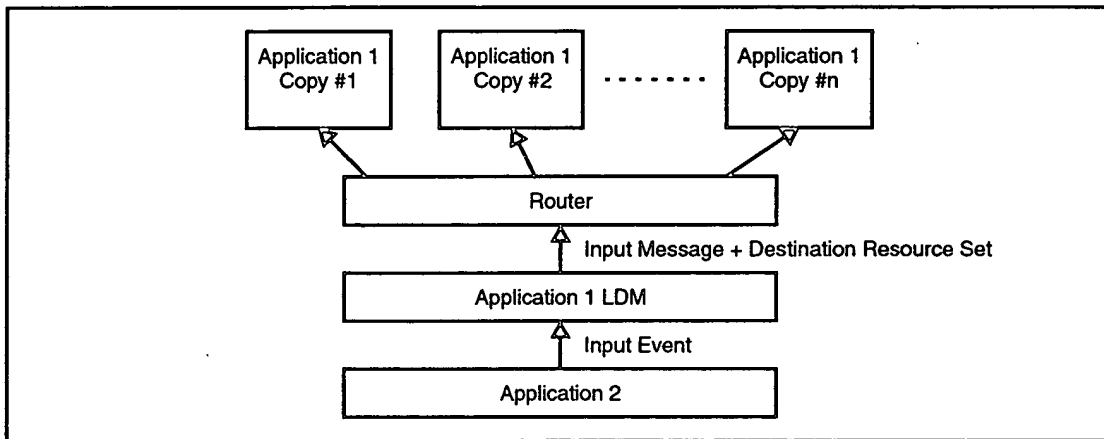


Figure 63

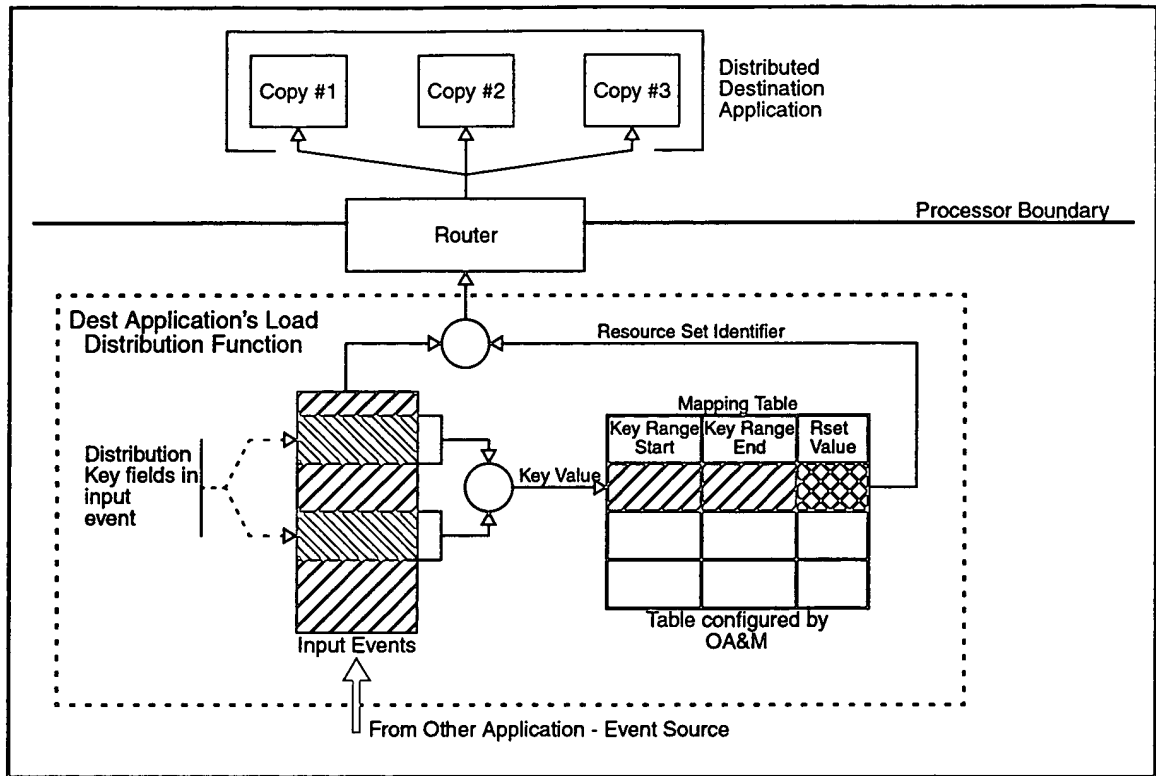


Figure 64

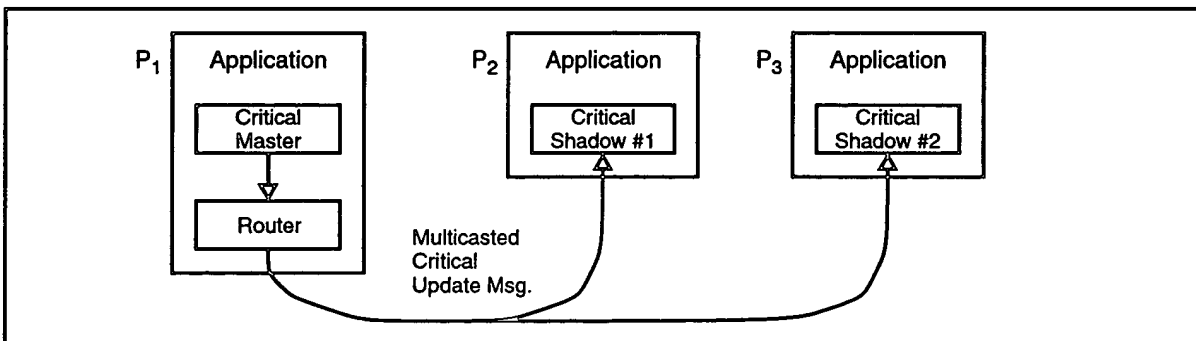


Figure 65

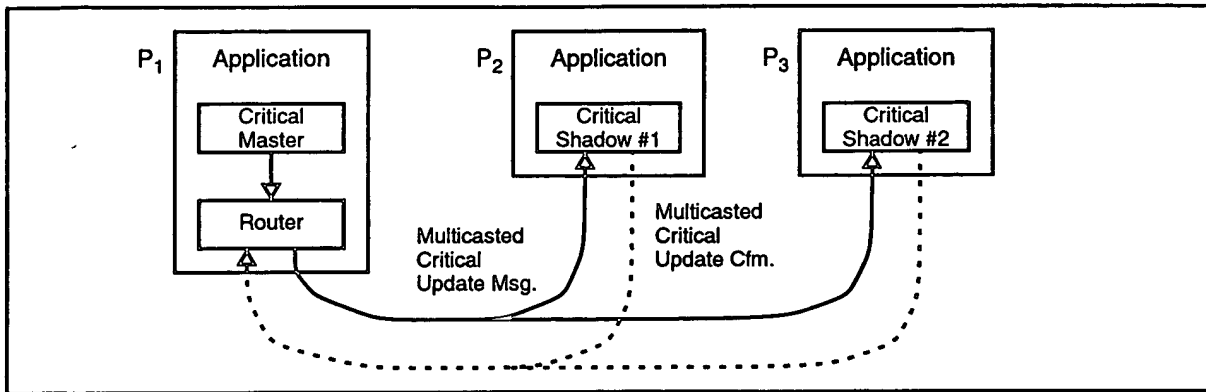


Figure 66

100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000

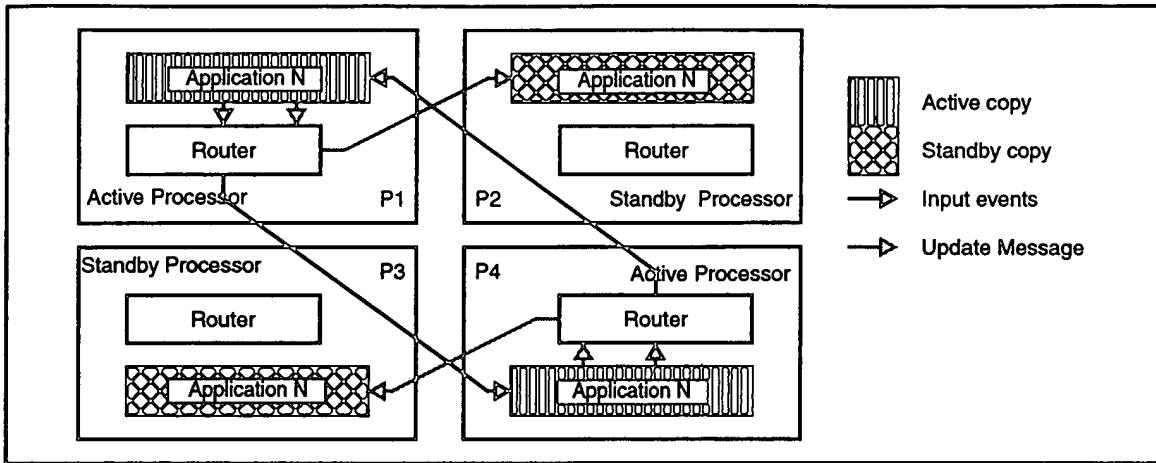


Figure 68